Transforming Bangladesh Agriculture: AI for Precision Crop Disease Management

by

Shahriar Hossain
24341118
Al-Zaber Nahin
20201058
Tasnuva Hassan
24341109
Zarif Ayman Haque
20201095
Nusrat Jahan Farin
21201826

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University

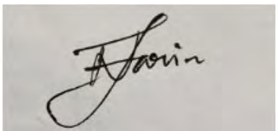
June 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

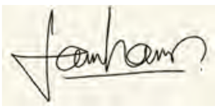


Nusrat Jahan Farin

21201826

Al-Zaber Nahin

20201058

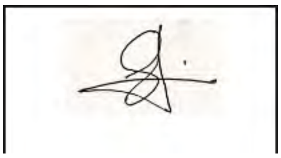


Tasnuva Hassan

24341109

Zarif Ayman Haque

20201095



Shahriar Hossain

24341118

Approval

The thesis/project titled "Transforming Bangladesh Agriculture: AI for Precision Crop Disease Management" submitted by

1. Shahriar Hossain (24341118)
2. Al-Zaber Nahin (20201058)
3. Tasnuva Hassan (24341109)
4. Zarif Ayman Haque (20201095)
5. Nusrat Jahan Farin (21201826)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 24th, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain

Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor; Chairperson
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Abstract

The agricultural sector encompasses a large chunk of the economy of Bangladesh as it has the necessary preconditions and factors to be suitable for agriculture. Agriculture is wholly at the whims of the environment and associated natural factors. Innovations from the time man has mastered the art of farming have allowed us to have in control some factors to ensure the desired output however there remains room for improvement and innovation especially in regards to disease detection. Currently even with a large agricultural sector, the methods for disease detection and risk management are lacking due to the inefficiencies in the system which can be very costly. To mitigate this technological innovations such as machine learning and image processing can be used to combat visible signs of disease and achieve early detection. In this paper we have explored the current options available and what can be done to make it suitable to our conditions, which ones are the best for our problem and finally we have proposed a solution we deem feasible. In our reviewed past works we have come across three models, namely Xception, VGG19 and ResNet50 which perform the best for our use cases, giving us the best results for leaf disease detection. These models have been implemented with a transfer learning approach to achieve the best results. Finally we have created a hybrid model approach combining Xception and a Vision Transformer to get the advantage of both a CNN and a Transformer to achieve the best result for our purpose.

Keywords: Crop diseases, Plant leaf diseases, AI-driven methods, Machine learning, Image Processing, CNN, ViT, Transformer, Xception, VGG19, ResNet50

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption. Secondly, to our thesis supervisor Professor Muhammad Iqbal Hossain sir for his invaluable advice and unwavering support for our work. Always available for us in times of importance. We would also like to thank all the different faculties who have offered us advice including but not limited to our defense judging panels and other departmental officers involved in our final approval processes. Finally we would like to thank our parents as they have made it possible for us to reach this goal. Their support and prayers are the foundation behind our success.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ANN Artificial Neural Networks

AUC Area Under Cover

CA Classified Accuracy

CNN Convolutional Neural Network

$D-CNN$ Deep Convolutional Neural Network

$GLCM$ Gray Level Co-Occurrence Matrix

HSV Hue, Saturation, Value

$k-NN$ K-Nearest Neighbours

LBP Local Binary Pattern

$LIME$ Local Interpretable Model-Agnostic Explanations

NB Naive Bayes

$R-CNN$ Region-based Convolutional Neural Network

RGB Red, Green and Blue

SVM Support Vector Machine

ViT Vision Transformer

Chapter 1

Introduction

1.1 Background

Worldwide, crop diseases are one of the main obstacles to agricultural productivity. Being a staple meal for about two thirds of the world's population, rice, wheat, and potatoes are the most prominent cereal crops that contribute significantly to global food security[30].Reduced food output brought on by diseases and pests that decimate crops is the source of food insecurity. Lack of research resources and knowledge makes it difficult to fully evaluate the effects of crop diseases on agricultural output in emerging and poor nations. This complicates, meantime, the assessment of crop disease impacts on agricultural productivity in underdeveloped and emerging countries. Food output is declining mostly due to harmful diseases, poor disease control, and rapid climatic changes[31]. Bangladesh is a very populous nation that mostly depends on its agriculture industry. Said to be the mainstay of its strong security are potatoes, rice, and wheat. But each year, major productivity losses are caused by crop diseases such late blight in potatoes, rust in wheat, and blast disease in rice. The most often used remedies for these are the environmentally damaging and often expensive use of insecticides.

Situated in South East Asia, Bangladesh has a typical hot-humid tropical climate where agricultural provides for 80% of the people. Being the most widely utilized cereal crop in Bangladesh, rice makes up 19.6% of the country's GDP and provides around one-seventeenth of its total revenue [30].

With 8.52 million hectares of arable land, this agriculturally focused nation grows a variety of crops like rice, wheat, potatoes, and so on. Still, Bangladesh's tropical environment encourages a high incidence of viruses and other infections. The strain of an ever-growing population is enormously putting pressure on food supply. These are the explanations of the viral disease outbreaks in many crops.

1.2 Problem Statement

As per a report[28], five main crop diseases are plaguing Bangladeshi farmers, perhaps lowering crop output by up to 62% yearly. To feed its people, Bangladesh, which ranks fourth in the world for rice production, yearly produces 39 million tons of rice, 1.1 million tons of wheat, and 9.61 million tons of potatoes. There are no proven pesticides for these infections, according to researchers; the only thing to do is take care. Moreover, the Bangladesh Rice Research Institute (BRRI) and experts state that the three main crop diseases in Bangladesh are wheat blast disease, rice tungro disease, and rice blast disease[28]. China was the first nation to record rice blast sickness, and later other nations. In Bangladesh, the blast disease outbreaks were documented during the boro season in 1980 and 1990. First, wheat became contaminated in Brazil in 1985. It had been a while before then, but in February 2016 wheat blast was first seen in Asia. This has a potential impact of up to 3 million hectares, which is severely restricting the global wheat output[30]. Besides, manually identifying the plant illnesses is really problematic. It takes an awful lot of work, knowledge of plant diseases, and a lot of time. For the identification of plant diseases, image processing and machine learning models can thus be used[19]. Our suggestion is to simplify the detection and preservation of crops for the farmers. From seedling to pre-maturity, crop disease can strike a crop at any point in its development. Main symptoms, however, show up on the leaves and nodes[12].

1.3 Research Objective

Our main objective is to create a model that can accurately identify the diseases in a plant and can also identify the types of diseases. This model might help the farmers to detect diseases easily and also help them to cure it. To achieve this we must:

- First, we will gather images of crops.
- clean and standardize the images for better comprehension
- Develop a program that will analyze the images and will detect if there is any disease and also will tell us the type of disease.
- Train the program with images of plants.
- Test the tool/model in a suitable environment.

1.4 Thesis Structure

In this paper, we used primarily 3 base models and 1 custom model to determine the detection of plant disease through image processing. In chapter 2 we studied various related works and formed a literature review. In chapter 3 we discussed our working plan and implementation strategy. In chapter 4 we discussed in depth the dataset used to detect diseases and the data processing techniques used in the datasets. In chapter 5 we learn more about the models used and the hybrid model that was build along with the results and analysis each model provided. Finally we concluded our paper in chapter 6 along with some challenges we faced while doing this research and also shared our future plan with this.

Chapter 2

Literature Review

2.1 Related Works

The topic of [2] was how manual observation is insufficient to identify the extent of crop damage that eventually results in a significant loss in the agricultural industry. As such, they put up a method that uses image processing and two convolution neural network (CNN) models, AlexNet and RestNet50, to identify and categorize plant leaf illnesses. This will enable the farmers to implement preventative actions. An picture dataset of potato and tomato leaves was gathered from Kaggle for this project. Images were processed using methods including acquisition preprocessing, restoration, segmentation augmentation, feature selection, and feature extraction once the signs of sick leaves were identified. Following the paper's suggestion, AlexNet and RestNet50 were used to categorize the photos. To further ease processing, RGB photos were also transformed to gray images utilizing color conversion methods. Noise reduction and improvement were also done during the data pre processing. 6000 distinct leaves were used to test the previously described CNN architectures, which categorized the whole dataset into groups of healthy and diseased leaves. Moreover, the same techniques were used to distinguish between the four distinct potato and tomato illnesses from the sick leaf photos. Such a system that would show the illnesses it discovered together with suitable preventative actions in a graphical style for users is presented in this study. It was shown that, following data pre-processing, training, and testing, the system outperforms AlexNet with 96.1% for leaf disease detection and 97% for classifying healthy and sick leaves. This also creates a space for misclassification, which can lead to a false diagnosis. One may also wonder if the graphical design can be effectively linked to a system that can give farmers useful information and close the gap between diagnosis and prevention.

[18] proposed a solution that classifies brassica crop's (cabbage, cauliflower, broccoli, kale) black rot disease based on the level of severity using vision transformer (ViT) model which is a Deep learning architecture [18]. This paper aimed to contribute to precision agriculture by introducing AI through the ViT model that will detect the infected leaves and assess the disease based on the level of infection. This research was conducted in two stages . The first one being developing a model that is robust , adaptive and reliable that will prevent yield losses . Then creating a proficient classification system to categorize diseases. A wide range of 5000 high

resolution images were collected and divided into training, validation and testing sets. After data pre-processing, multiple augmentation strategies were applied to the training set to make the model more resilient. The validation set, which is a subset of the training set, was used for hyperparameter tuning (learning rate, batch size , optimization techniques) that can prevent overfitting. Moreover, the paper goes into describing how CNN analyzes images in a grid whereas the ViT model treats images as sequences of tokens which is why in depth examination is possible by imposing self attention techniques. To evaluate the workability of the ViT model, comparative studies were carried out . DL algorithms such as PyTorch, tensor flow were utilized along with Gradient- weighted Class Activation Mapping algorithm to improve the interpretability of the model. In addition to that, Cohen's Upsilon, a performance matrix, was used to evaluate the model's prediction for the categorization labels. The accuracy of the ViT model was concluded to 96.17% with a Cohen upsilon value of 0.94 that suggests the model's prediction aligns well with the observed labels. Despite such precision , ongoing challenges such as - misclassification, disease progression and symptomatology still persist which needs to be addressed in future research.

Using both healthy and sick crop photos, a study by [26] used six CNN models—Xception, VGG16, ResNet152V2, InceptionResNetV2, DenseNet201, and MobileNetV2—to diagnose crop illness.The four crops under consideration are rice, potatoes, corn/maize, and wheat. This paper compares CNN models using grayscale and colorful pictures. Following dataset collection, many pre-processing methods were used with the support of the Keras and Tensorflow backends. The data was divided into 80:20 ratio for the training and testing sets and scaled to 331×331 pixels. Next came the sophisticated DL models. Moreover, many layers were imposed for improved accuracy: Batch Normalization layer, Global Average Pooling 2D, Dense(512, activation='relu'), Dense(256, activation='relu'), Dropout(0.5), Dense(128, activation='relu'), Dense(120, activation='softmax'), Dense(4, activation='softmax'). The results showed that, while colorful pictures perform better and provide a greater accuracy, gray images occasionally provide a little accuracy. Compared to MobileNetV2, for example, grayscale corn photos perform better with ResNet152V2, InceptionResNetV2, Xception, and DenseNet201. The validation study shows that MobileNetV2 has the best accuracy of 94.39% and DenseNet201 the lowest accuracy of 77.42% for colored pictures. Still, DenseNest201 had the greatest grade of 95.52% for grayscale photos and the lowest of 75.48%. This time, grayscale did better than colored photos. Moreover, compared to grayscale photos (92.57%), the colored images for potatoes provide the best accuracy (96.34%) for all six models. Additionally included in the work was the SAM (Successive Approximation technique) approach for generalization enhancement. For corn and wheat, this study achieved a greater accuracy (over 92%), for potato leaves a moderate accuracy, and for rice leaves a lowest accuracy. One drawback is that the study is limited to examining certain crop species; so, not all 15 crop species have had their CNN models evaluated. It also proposed the development of a web-based interface for a comparative hybrid model.

[37] studied potato leaf disorders that are usually brought on by microbes, infectious agents (fungus, viruses, and bacteria), or any genetic abnormality. Since deep

learning models map functions more well, they were suggested for the detection of leaf diseases. Consequently, a superior feature generator than conventional image processing models like LBP. Two categories separated the carefully selected dataset that was gathered from Kaggle. One with and without a backdrop. Three categories of potato photos are distinguished: healthy leaf, late blight, and early blight. In this work, an orange data mining tool (open source Python library) using a CNN based method extracted and classified features. The work suggested using CNN architectures VGG19, VGG16, and InceptionV3 for feature extraction. Among them, VGG19 outperformed as the model is pretrained on a high resolution ImageNet dataset and the only pre-processing required to center the data is to remove the mean RGB value from each pixel. Statistical analysis benefits from centered data coverages, which complete faster. Moreover, the retrieved characteristics were run through classifiers including logistic regression, SVM, KNN, and neural network. Among all the previously described classifiers, logistic regression produced a better accuracy of 97.7%.

[14] published a conference paper for detecting leaf disorder by using image processing methods [14]. Disease can be detected by any distorted images also within a short period of time. So it is easier for farmers to detect disease without having knowledge using modern softwares. 13000 images of different types of healthy leaves and disease affected were collected from various sources like github and google. From these, 10,000 valid images were splitted into 80% training and 20% testing sets. The model is based on CNN with numerous layers, and features Relu activation functions with different filter sizes, kernel sizes, padding, and strides settings in each layer. This architecture also includes a final dense layer with softmax activation , convolution layer and max pooling. Adam '3' optimization algorithm was used in a model with a learning rate of 0.001. An understanding of how the CNN analyzes and alters input pictures is provided by the model's visualization of incremental image changes over several layers.

[26] worked on one healthy leaf class of paddy and four varieties of paddy leaf diseases [13]. This work attempts to enhance the results of paddy leaf disease identification by automated detection with deep learning CNN models, which may reach the maximum accuracy. The procedure they followed supersedes the traditional manual illness detection procedure, which takes a lot longer and has dubious accuracy. Deep convolutional neural network (deep CNN) is a kind of feed-forward neural network used to alter network parameters to reduce the cost function value. The work of these models has been very beneficial to the analysis of visual imaging, object identification, image or video classification, and natural language processing. More layers are present in the convolutional neural networks supporting the deep CNN model than in the traditional CNN model. Many times, models contain convolutional, activation, pooling, flatten, dropout, batch normalization, and dense layers. An overfitting is avoided by using the dropout. Activation functions like ReLu and Softmax are introduced to network layers to help find intricate patterns in data. Four models were used: VGG-19, ResNet-101, Inception-Resnet-V2, and Xception. Studying these algorithms, researchers found that Inception-Resnet-V2 surpassed all other algorithms in terms of precision, recall, and F1 score in addition to achieving the highest accuracy of 0.9286. After Inception-ResNet-V2, Resnet-101

has shown an accuracy of 0.9152. The Xception model has an accuracy of 0.8942 and the VGG-19 model the lowest, at 0.8143. Not only does accuracy receive the lowest marks, but also precision, recall, and F1 Score. The epochs of the training techniques total 100.

In the paper [21], published in 2020, on developing a system for detection and identification of leaf diseases the authors explore improvements they could make in the detection, specifically comparing two segmentation methods which are K-means clustering and color thresholding. Further on it evaluates two feature extraction algorithms for classification of images along with the aid of an SVM classifier, these are GLCM and CNN, specifically ResNet-50. The authors elaborate the reason behind comparing the two segmentation approaches is to determine the better segmentation method for segmenting the regions of interest. The dataset, derived from the PlantVillage repository, contains images of tomato leaves with four classes: Septoria leaf spot, Alternaria, Bacteria disease and finally healthy specimens. Further processing on the dataset includes resizing and contrast enhancement for better distinguishment of features. Segmentation is then performed using the two methods discussed earlier and first there is the K - means algorithm. The algorithm, which categorizes pixels uses similarities in RGB color values and creates a cluster, with the 'K' in K - means , additionally also using help from spatial and texture features with the help of a Gabor filter to aid performance. A initial value of k = 2, which determines the number of different clusters the image is segmented, is used but the result is not satisfactory, with texture and spatial data added there are some improvements but even with higher cluster values of 3, 4 and 5 there are not much improvements and often resulted in the background being clustered with foreground leaf details. The next segmentation algorithm used is color thresholding, for which the images are converted from the RGB to HSV colorspace for better perception. This yielded better results with it successfully eliminating the background pixels, a small drawback being it also eliminated some pixels in the region of interest but this was rectified using a binary mask where it would remove the resulting black spots in middle of the leaf and connect the edges. The authors derived color segmentation to be a better method compared to K-Means on the basis of the former being able to remove background color details more effectively. Heading into feature extraction and classification both of the mentioned feature extraction methods yield good results with an accuracy of greater than 80% for both overall. Looking into the results further, the authors have shown the CNN approach to be better with an accuracy of 96.63% on average, higher than the GLCM approach which attained an accuracy of 82.63%. They have chosen to use the CNN along with the SVM classifier as the best approach. The authors also have developed an interactive system which receives a dataset to train from, then classify images it receives to identify the correct diseases affecting it. In the concluding statements the authors have stated the interactive system can be made to work with many other left diseases with proper training and tweaks to the system. They further suggest exploring Deep CNN as a means to better segmentation of images and to gain the ability to implement more complex datasets.

Next we have paper [24], which follows a similar approach to paper [21]. They also follow a deep learning approach, particularly CNNs which backs up the claims of the previous paper. The authors of the paper have come to this conclusion based on studies done by [20] who use a hybrid model based on CNNs to classify features. They also analyze further studies done by [6] who use AlexNet and SqueezeNet CNN architectures to detect tomato leaf diseases. Going further we can see the approach they used for their study where the data set they are using contains 35,000 images from the Plant Village repository as was also used in the previous study [21], and they mainly use apple, corn, grapes, potato, sugarcane and tomato. The images were then preprocessed for use by the classifier with further augmentations done such as rotation of images by 25 degrees, flipping or shifting them horizontally and vertically to increase diversity of the dataset and ensure robust training. The CNN model used the 35,000 images for training and 1,000 on field images were used for testing. The CNN model incorporated an Adam optimizer and training was done across 75 epochs at a batch size of 32. For results they achieved a 100% accuracy in recognizing healthy corn leaves and those affected with gray leaf spot disease. Tomato late blight diseases with a 99.62% accuracy and early blight with 75.36% accuracy. Grape leaves affected by late blight and black rot were identified with an accuracy of 100% and 95.09% respectively and finally Apple leaves with black rot diseases were identified with a 100% accuracy and 99.99% accuracy for healthy leaves. It can be seen that CNN based models are great from plant disease recognition and detection with high accuracy rates. The authors suggest future improvements in the form of expanded datasets, using different CNN architectures and optimizing parameters for the models to improve performance.

Our subsequent work [23] focuses on developing an explainable AI implementation and transfer learning framework. Three separate pre-trained models were employed by the authors for this work. Their study covers 38 distinct illnesses in 14 different plant species. They have moreover demonstrated the interpretability of the models' decisions using the explainable AI package Local Interpretable Model-Agnostic Explanations (LIME). They arrived at this result by examining other research, like study [3] in which the authors employed a neural network technique to achieve a 93.67% accuracy. In study [22] deep learning methods like CNNs have been used to obtain 98.96% accuracy with ResNet and a custom dataset of 500 plant photos; in study [1] MobileNet-v2 is used to reach 99.7% accuracy. More sophisticated Deep Learning techniques are examined in studies like [36] in which the authors employed DL VGG-19 to achieve an accuracy of 87.4% in addition to Inception-v3, VGG-16, and VGG-19 with differing degrees of success. Regarding the others, Inception-v3 and VGG-16 each obtained 89% accuracy, the highest among the other deep learning techniques. An other paper [25] described the use of GoogleNet, Resnet50, and VGG16 models, parsing them using SVM and KNN models. With 97.82% percent accuracy, VGG16 achieved the highest promising degree of accuracy. According to their findings, SVM method is the most effective one for leaf disease diagnosis. Proceeding, the authors report that they obtained their dataset from Kaggle, preprocessed the images (resized from 256x256 pixels to 128x128 pixels), normalized the pixel values by dividing with 255, and divided the images into three sets: 43,429 for training, 5,417 for validation, and 5,459 for testing. To deal with the data constraint, the photos were additionally enhanced. For the current work, three pre-trained CNN models—EfficientNetV2L, MobileNetV2, and ResNet152V2—were applied. The models contain pretrained weights from ImageNet. We have applied

Adam optimiser with 0.0001 learning rate. A multiclass classification job has led to the implementation of a categorical cross entropy as loss function and the existence of a softmax activation function in the 38-neuron output layer. Furthermore having the greatest F1 score, EfficientNetV2L clearly performed the best out of all the models. Proceeding, the authors have also put into practice a XAI framework, more precisely LIME, which displays the interpretations on the areas utilized for the EfficientNetV2L prediction. The authors outline their constraints in their closing notes, including data and infrastructure resources. They also intend to use regional characteristics of plants and illnesses in future work to provide more sophisticated forecasts.

Our next paper [39], the authors also explore a CNN approach based on studies they have reviewed. Such as a study [9] they reviewed showed a deep CNN comprising of 9 layers used to detect multiple diseases with an accuracy of 96%. Study [27] proposed changes with a deep CNN model comprising of 14 layers. The authors reviewing the studies acknowledge the computational challenges with using such advanced models even if they provide a high level of accuracy. It can not be feasible to use these algorithms at the field level, the goal of the authors is to implement a lightweight deep CNN model to diagnose Apple plant leaf diseases.In a paper by [15] they used region-based convolutional neural network (RCNN) for the purpose of detecting apple diseases diseases with a great level of accuracy. Further explored is the possibility of transfer learning and fine tuning such as in a study [8] where they used CaffeNet, AlexNet, VGG and GoogLeNet for plant disease identification and Coulibaly et al. using a VGG-16 model, which they fine tuned their purposes, for millet crop mildew diseases. In another study [17] they have found the use of AlexNet in the prediction of apple scab diseases, however the resulting prediction accuracy was not significantly better than current methods. The authors have implemented a deep CNN (Conv-3 DCNN) model for this scenario. The dataset they are using is the PlantVillage dataset containing 3171 RGB images of apple plant leaves and these are separated into four classes, black rot, scab, apple cedar rust and healthy leaves. The deep CNN model they have used, uses 3 convolutional layers to create feature maps from filtered information which is then fed to 2 fully connected layers and after that into 3 max-pooling units. The model has been assigned 0.0001 as the learning rate and the mini batch size set to 50, and then trained for 1000 epochs. A ReLU nonlinear activation function is implemented for the convolution layers. For the first convolution level they have used 32 filters (3×3) pairing with a padding and stride of 1 and 1 respectively are selected to convolute over the RGB images of size 256x256. This gives us 32 feature maps of size 254x254. At the first pooling layer, the resulting feature maps are first downsampled by a kernel of 2x2 and 32 feature maps of size 127x127 are created. The dataset is split into training, validation, and test sets comprising 228, 634, and 319 images respectively. Upon testing, the model achieved a classification accuracy of 98%. Further results showed the model required overall less computational resources compared to other deep CNN models, suitable for the authors goal of deploying it on handheld devices. The authors suggest a more detailed dataset, with examples from various geographic areas, different image quality and variability.

For the paper [11] the authors explore the possibility of using a lightweight deep

CNN model to obtain high level hidden feature extraction. They explore the possibility of reducing the training parameters as well as the iteration times as compared to common transfer learning models, namely GoogleNet, AlexNet and VGG16. The proposed model concatenates deep features along with the traditional handcrafted features and textures (LBP). The dataset they use is the PlantVillage dataset containing 3171 images. The dataset comprises the following categories of images with Scab which includes 630 images, followed by Black Rot comprising 621 images, then Cedar Rust with 275 images, and Healthy with 1645 images. Data augmentation is done to balance the different classes to bring the total to 4645 images. The dataset is split in 80% training and 20% validation. The tomato leaf dataset was also containing 18160 images, grape leaf dataset containing 4062 images and both these datasets were augmented as needed and split into training, validation and testing. Images are pre-processed using techniques such as filtering and sharpening and then filtering. The images are resized to 64x64 pixels before being fed into the model. There are three convolutional layers, each with a kernel size of 3x3. There are 32 filters in the first layer, increasing to 128 in the last convolutional layer. A max-pooling layer with a kernel size of 2x2 is then applied. For the activation function, ReLU is used. As discussed earlier they directly concatenate the extracted LBP features with deep features in the flattened layer. For the results the proposed CNN model achieves a high accuracy, for the Apple dataset it achieves a 99.9% training and 99% validation accuracy and for the Tomato leaf data it achieves a 99.4% training and 96.6% validation accuracy. A 98.8% and 96.5% test accuracy is achieved for the two datasets respectively. The authors also highlighted the increased accuracy their proposed CNN model had compared to transfer learning based approaches such as VGG16 and AlexNet, achieving a higher precision, recall, F1 and accuracy score in all classification scenarios. In the concluding remarks the authors suggest exploring LBP variants further as they provide a texture feature extraction for additional benefits. They also suggest enhancing the proposed model and expanding its application in the agricultural sector.

The use of AI has increasingly gained attention in agriculture, particularly for its ability to improve the early detection of plant diseases. In the study [33], the authors have used convolutional neural networks, a deep learning-based model, to accurately identify plant diseases from leaf images. Their work begins with data collection and preprocessing, which includes resizing and normalizing plant leaf photos sourced from online repositories. These images are then standardized by resizing them to 256×256 pixels and normalizing their pixel values, typically by dividing each pixel value by 255 [33]. To improve generalization, data augmentation techniques are also applied, simulating variations in the dataset. The dataset is divided into subgroups for training, validation, and testing in an 80:10:10 ratio [33]. Additionally, the CNN architecture developed in this study consists of several convolutional layers with various filter sizes, followed by pooling layers, dropout layers to prevent overfitting, and fully connected layers. A softmax activation function is used in the output layer to classify disease types. Here, the cross-entropy loss function and Adam optimizer are used for training the model, which lasts for 50 epochs with early stopping to prevent overfitting. This approach produces strong results with an overall accuracy of 92%, precision of up to 95%, recall of up to 96%, and F1 scores that are constantly around 91% [33]. The authors also use confusion matrices and accuracy/loss curves to analyze performance and ensure the model is well generalized rather than overfitted. Beyond this specific study, several past studies have shown that CNNs are quite accurate at detecting plant diseases. For instance, in the paper 4, they revealed that the CNN model could achieve very high accuracy, up to 99.35%, to detect 26 diseases on 14 crop species. Likewise, it is shown that 5 CNNs trained on large agricultural image datasets could reach an accuracy rate of over 99.53% in 25 plant species to recognize 58 diseases. Another key approach is transfer learning, which involves adapting existing models to new but related tasks. This method is efficient for improving accuracy while reducing the training time. As noted in 6, fine-tuning pretrained networks can significantly enhance the performance of plant disease classification systems. Moreover, researchers have also explored how CNNs can be applied in real-time and mobile platforms to enable instant disease recognition in the field. The authors 7 designed portable AI systems that deliver high performance even on mobile devices, making it easier for farmers to diagnose diseases on-site.

2.2 Models Definiton

2.2.1 VGG19

Deep Convolutional Neural Network includes VGG19 which has 19 layers and is intended for image classification tasks. To minimize spatial dimensions, it uses numerous stacks of tiny 3x3 filters with max-pooling layers. It consists of three fully connected layers, sixteen convolution layers, and a softmax for classification at the end. VGG19, even being a simple architecture, is a popular choice for applications like image recognition and transfer learning due to its robust performance and ease of use.

2.2.2 Xception

Extreme Inception , also known as Xception, is a deep learning model built upon Inception architecture by utilizing convolution neural networks . Inception modules use multiple filters including the pooling layers with a reasonable number of parameters that balances the complexity . Whereas in Xception, these modules are replaced with depthwise separable convolution layers . As a result, the number of computations and parameters are reduced significantly. Hence , the efficiency can surpass the performance of other models such as VGG and InceptionV3 especially on large datasets . Moreover, by facilitating transfer learning, Xception allows fine-tuning that can increase the accuracy and give better results , especially for smaller datasets.

2.2.3 ResNet50

ResNet50 belongs to the ResNet clan of deep-convolutional-networks and tackles the vanishing-gradient headache that makes very-deep models hard to train. It swaps ordinary stacked layers for residual blocks, adding shortcut paths that let the net copy learned features straight to later layers. These skip links keep gradients flowing smoothly, so architects can safely pile on more layers without speed or accuracy crashes. Inside ResNet50, fifty layers are grouped into convolutional and identity blocks that together refine every incoming feature map. The layout extracts subtle image patterns while keeping memory and compute demands reasonable for many real-world devices. Because it is reliable and quick, ResNet50 shows up everywhere-from research notebooks to factory cameras-checking pictures of tumors, trucks, and rotting fruit.

2.3 Background Study

2.3.1 Convolutional Neural Network (CNN)

CNNs are a type of deep learning model mainly used for image analysis. They use layers of filters to automatically detect visual patterns like edges or shapes. By combining convolution, pooling, and dense layers, CNNs can learn complex image features. Therefore, they are widely applied in tasks such as object recognition and classification.

2.3.2 Vision Transformers (ViT)

Unlike traditional CNNs, Vision Transformers apply transformer architecture to visual data. They break an image into small patches and treat these like a sequence of words in text. This allows the model to capture global relationships across the image more effectively. As a result, ViTs often require large datasets but offer strong performance when scaled.

2.3.3 Transfer Learning

Transfer learning involves taking a model already trained on one task and adapting it to another. Typically, earlier layers are kept intact while the final layers are modified for a new goal. This approach significantly reduces training time and is ideal for smaller datasets. Consequently, it has become a popular strategy in fields like image and language processing.

2.3.4 Hybrid Models

Hybrid models combine different types of architectures to enhance overall performance. For example, one might use CNNs for local feature extraction and transformers for capturing long-range dependencies. By leveraging the strengths of multiple models, they often achieve better accuracy and robustness. Thus, hybrid approaches are gaining popularity in complex AI tasks.

2.3.5 Classification Head

The classification head is the final segment of a neural network that outputs predictions. It usually consists of fully connected layers followed by a softmax function to assign probabilities. This component translates the learned features into specific class labels. Notably, it can be replaced or fine-tuned during transfer learning for different applications.

2.3.6 Adam Optimizer

The Adam optimizer is a commonly used method for updating neural network weights during training. It smartly combines the benefits of momentum and adaptive learning rates for efficient convergence. Thanks to its adaptability, it performs well on noisy or sparse datasets. As a result, it's often the default choice in many deep learning frameworks.

2.3.7 Learning Rate

The learning rate determines how much the model adjusts during each training step. If set too high, the model might skip over optimal solutions; if too low, training may be slow. Because it greatly affects performance, tuning it carefully is crucial. To address this, techniques like learning rate schedules or warm restarts are often used.

Chapter 3

Methodology

3.1 Working Plan

Reviewing past works we have attempted to see what the state of the art solutions are to our problem. Once we have determined the state of the art solution to our leaf disease classification task we have tested the solutions on our dataset that we have collected from recent sources. After testing, we have chosen the best performing model with which we have improved further upon by evolving it into a hybrid model to act as a viable solution to our problem.

1. Literature Review:

- Model Identification: Identify the best performing models and if they are the state of the art.
- Dataset Identification: Evaluate wether the dataset used matches our real world conditions for the model to viable for our purpose.

2. Dataset Collection:

Crucially find a dataset meeting our following criteria in order to build a robust model.

- Sample Size: The dataset must have an adequate number of samples for our model to generalize upon effectively.
- Real World Conditions: The dataset images must represent on the field conditions.
- Recency: The dataset must be recent in order for it to have the level of quality expected from current technology.

3. Transfer Learning Approach:

Based upon review of our of previous works we will implement the models using a pre trained transfer learning approach.

4. Build Hybrid Model:

Upon review of state of the art model testing we will attempt to create a model which does the following.

- Good Generalization: Effectively generalize well upon our real world dataset.
- Good Performance: Provide us with efficient performance while also being effective on our data.
- Scale: Model should be scalable for future improvements.

3.2 Implementation Strategy

Our implementation strategy can be summarized in the following points.

- Data Collection: Collect our required data, create data for testing, training and validation and prepare them for our models to train upon.
- Data Processing: Process our data for our models to use, applying image augmentation, generating class weights and finally creating the correct input size and normalization.
- Training the Models: Train the weights of our model by running them for a number of epochs on our dataset until we get an acceptable result from each.
- Build Upon the Models: Determine which model performs best upon our training runs and testing and select that model to be improved into a hybrid model approach for our purpose.

3.3 Innovations and Benefits

Hybrid Model: Implementing a robust and versatile hybrid model combining ViT and Xception to provide us good generalization performance on our data while performing exceptionally.

Robust Assesment: This model can perform tasks that other models, built upon ResNet, VGG and other architectures, cannot with their limited generalization performance using lab sampled datasets cleaned and polished for noise and other artifcating.

Good Performance on Data: The model should have good performance on our real world data which do not conform to ideal conditions on the field.

Portable: The model can best used portably, possibly offline for on the fly disease classification.

Improve Agricultural Yield: The model can provide timely identification of diseases using CNN and ViT hybrid classification mechanism without the need for human intervention other than spotting visible signs of disease. Timely intervention can be made based upon model's assessment.

Model Scalability: The model can be scaled to wider implementation, trained to identify a wide variety of leaf diseases, combined with other models possibly for considering more real world metrics for better disease identification.

3.4 Workflow Visualization

A flowchart is depicted in Figure 3.1 which shows how our models are trained to get our desired output. The dataset is first processed for use, then loaded into training our models which upon training the best weights are saved as well as the training data is preserved for our assessment. Finally analysis is done of the final models to determine the best performing one.

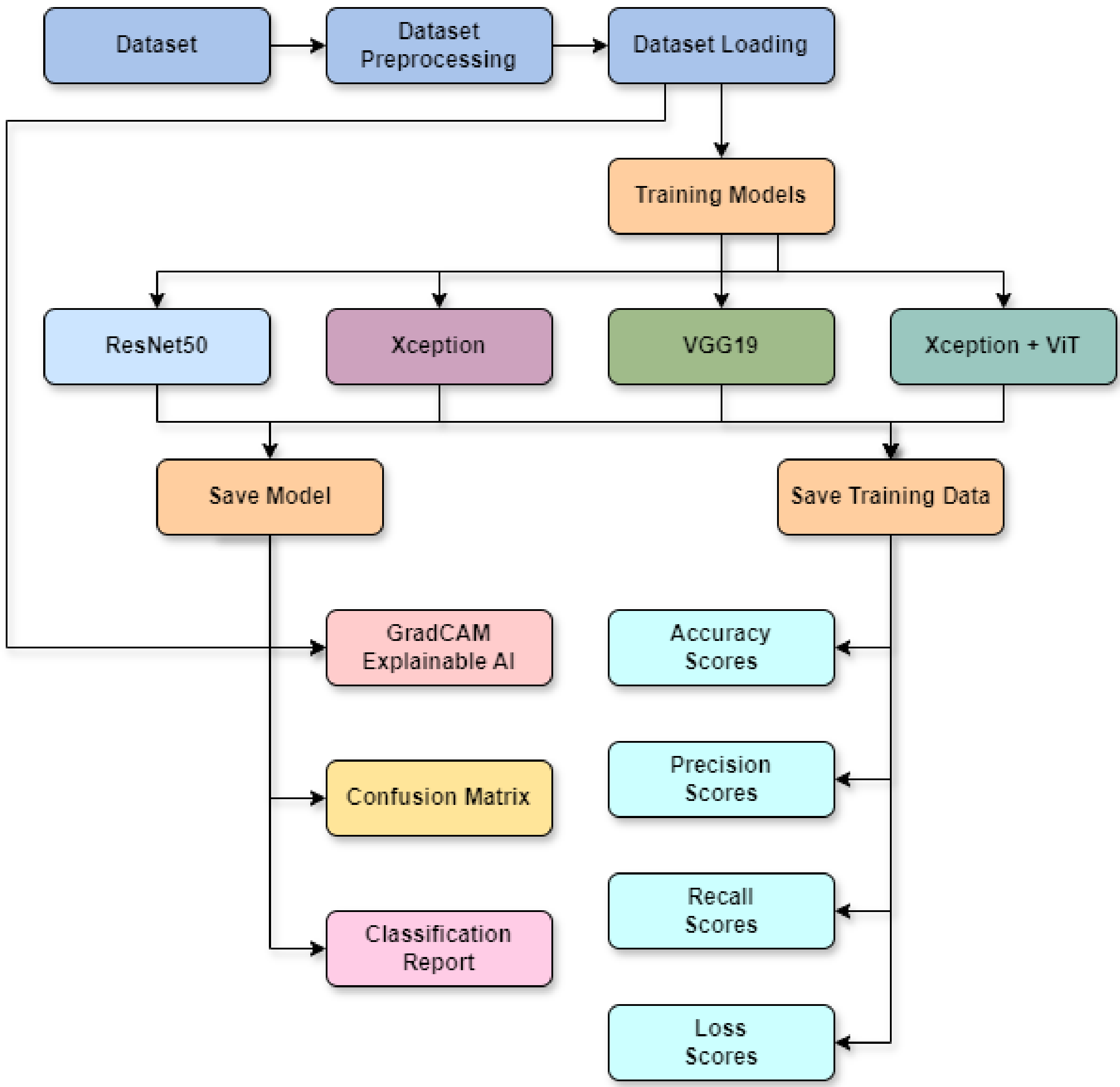


Figure 3.1: Flowchart depicting the work plan.

Chapter 4

Dataset Overview and Data Processing

4.1 Dataset Overview

As part of our training process we have used 3 datasets for our crop types: Rice, Potato and Wheat. The focus of this training process is to identify various visible leaf diseases and so the datasets contain images of leaves of the aforementioned crops which showed visible signs of different diseases affecting the crop types. There is an abundance of datasets meeting our criteria especially on popular research and online journal sites and for our purposes we have picked our datasets from Kaggle as we found the datasets being suitable to our use case and have key traits such as recency of the dataset and sample size. All the datasets have been published fairly recently and contain high resolution imagery on par with what could be captured using modern mobile device cameras.

Taking a look at our datasets; first we have the dataset for rice leaf images- Rice Leaf Diseases Detection by[29] which contains 30,000 images of diseased rice leaves, the dataset contains a mix of augmented and actual images In our training we have decided to use the train and test subsets under the regular images for training and validation respectively. The augmented images have been reserved to be used for testing purposes. Moving on, the dataset contains 10 classes of leaf images as follows: Bacterial Leaf Blight, Brown Spot, Healthy, Leaf Blast, Leaf Scald, Narrow Brown Spot, Neck Blast, Rice Hipsa, Sheath Blight and finally Tungro. A detailed breakdown of the dataset train and test subsets can be observed in Figure 4.1.

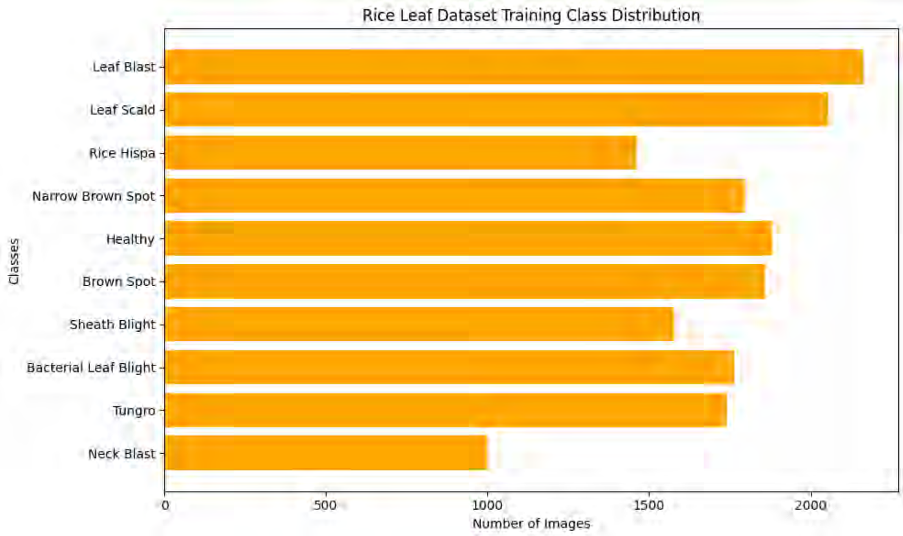


Figure 4.1: Distribution of our Rice Leaf Dataset for Training.

Our next dataset is for potato leaf images- Potato Disease Leaf Dataset(PLD) by[32] which has 4072 files and 3 classes of leaves, Early, Late Blight and finally Healthy. The dataset has a smaller sample size than our previous dataset but it follows our requirements of being clear, real world use case scenario images. The dataset comes pre-divided into train, test and validation subsets and we have used them as arranged by the authors. A detailed breakdown of the dataset train and test subsets can be observed in Figure 4.2.

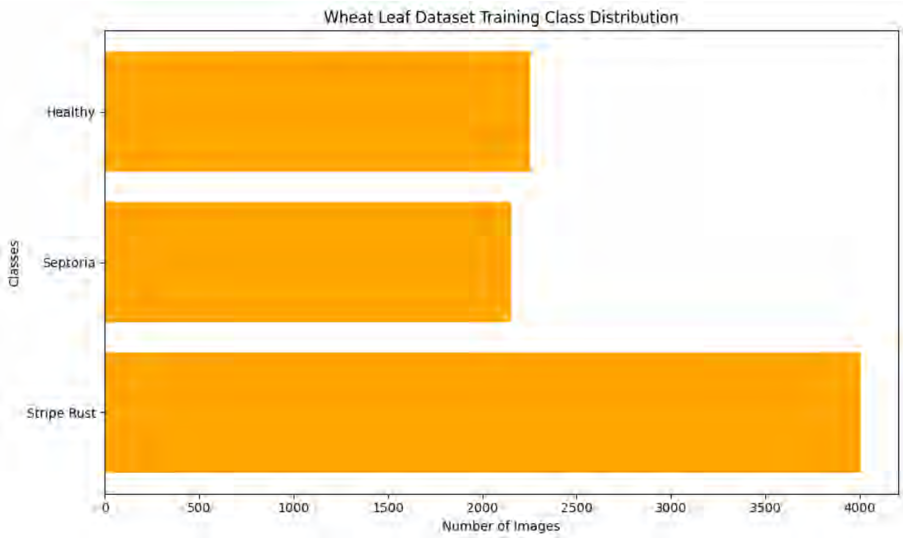


Figure 4.2: Distribution of our Potato Leaf Dataset for Training.

The last dataset we have used for training is[7] which has a little over 8000 files overall. This dataset performed the best in our training due to having the best samples in terms of quality and meeting our criteria. The dataset also underwent more augmentations which is discussed later on which may have helped with training.

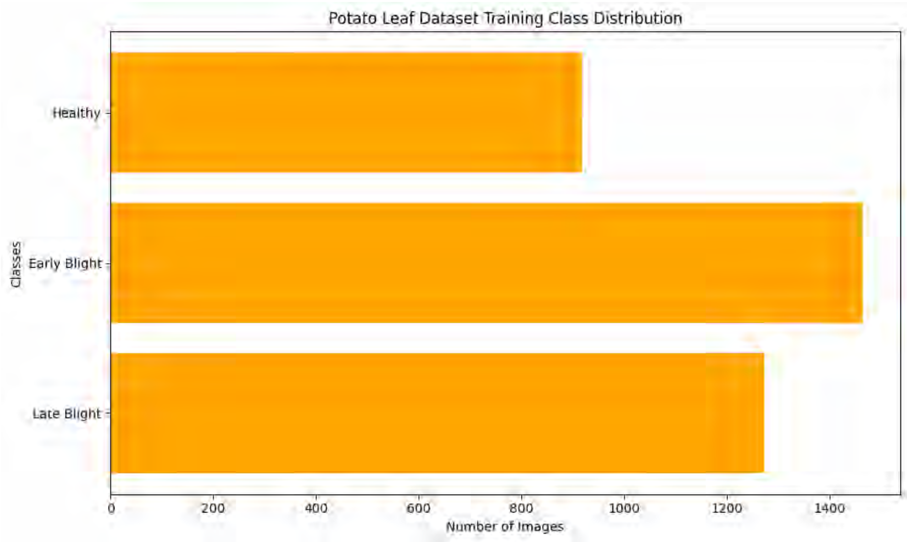


Figure 4.3: Distribution of our Wheat Leaf Dataset for Training.

4.2 Data Processing

The datasets we have acquired meet our criteria and have some level of processing done beforehand by the authors, even so it requires some level of pre-processing in order to provide proper conditions for training and so the aforementioned datasets have undergone a series of changes, calculations and adjustments to create ideal conditions for our models to generalize the data better.

The first step we took was fixing mislabelling and organizing data for training. The dataset for Rice had mislabelled classes for the "test" set which we have used for validation, inconsistent with the train set. To resolve this we have created a second folder and remapped and renamed the directories from the original set with the correct labels and copied it to the aforementioned folder. This allowed us to have our train and validation set for rice leaves, as for the test set we have used a set of the augmented half of this dataset we did not use for training to ensure we have a good batch of unseen data. Next, we have the potato dataset which we have left as is, there was no need for any of the first steps for it. Lastly, we have our wheat dataset which came as a single set. So to get our train, test and validation sets we have split this into a new directory like earlier with 80% for train and 10% for test and validation each. This concludes our dataset structure handling. The next step we took was calculating class weights for our training sets as the classes in our datasets had varying sample sizes i.e imbalanced. The class weights assigned to each of our classes can be seen below. The class weight helps adjust the loss function to ensure rare classes get greater importance while more abundant classes get lower importance by adjusting the penalty for each instance of misclassification.

Dataset structure handling and accounting aside now we move on to image pre-processing steps, here we have three common preprocessing steps done to all sets. First off, we have resized images to 224x224 to match our model input size and also have normalized the pixel values using rescaling using a factor of 1./255, normalizing values from 0-255 to a 0-1 range. Next we have done one-hot encoding for our labels. All staples for image classification tasks. After this we have done more image augmentations for our training sets using the ImageDataGenerator function available from the Tensorflow Keras API. The augmentations for our datasets vary from dataset to dataset and for our rice and potato leaf datasets we saw enough variations in our samples for these two to apply moderate augmentations, namely a horizontal flip, zoom range of 0.2 and rotation of 10 degrees. These were our first datasets and we have kept it simple for them, however for our wheat leaf dataset we have applied more diverse augmentations. The horizontal flip was kept, with zoom range and rotation reduced to 0.1 and 5 degrees respectively, accommodating more augmentation parameters like a width and height shift of 5%, brightness variations from 0.9 to 1.1 and to account for the shifting using fill to nearest. These augmentations for our images help our model see more variations, boosting generalization and performance.

Lastly we have loaded our data for our models to use, applying a shuffle to our training data so that when data is fed to the model during training it helps eliminate learning bias and create more variance to avoid overfit. The data is fed using batch size 32 for our rice and wheat data whereas for potato dataset we had used 64 due to constraints we were facing in terms of raw dataset size in terms of memory used and the time it took to load data and train with our limited time allowance on our training hardware.

Chapter 5

Model Architecture and Result Analysis

5.1 Model Architecture

Going off of our reviews of previous works we have determined three models that tend to have the highest accuracy for leaf disease classification tasks. As such we have used those models for our preliminary testing to determine the best approach for our purpose.

5.1.1 Xception

The first model we have used is based on the Xception architecture where we leverage transfer learning to get the best performance. Talking about the architecture, the name stands for Extreme Inception and it uses separable convolutions which split standard convolution into smaller steps which helps reduce parameters and computations while maintaining performance. It can be broken down into three flows, the Entry flow for extracting low level features, middle flow using separable convolution blocks for complex patterns and finally an exit flow which processes these features for use by the classification head. The advantage of transfer learning by using the pre-trained Xception model, which has been pretrained on the ImageNet dataset, is that it helps us leverage previously learned visual patterns by freezing the layers of the core Xception model and using the pre assigned weights [4]. On top of this we use our own classification head which uses the feature map derived from the deep layers of the Xception feature extractor to give us our required predictions. The top layer or classification head we have used in this case is a GlobalAveragePooling2D layer to flatten dimensions and compress spatial information, which is then fed into a Batch Normalization layer which acts to normalize the inputs to layers to speed up training. This is then used with a Dropout layer which serves to prevent overfitting by randomly dropping neurons, in our case 20% of them, and this helps with generalization. Finally there is a dense layer which outputs the final predictions for the 10 or 3 classes using the feature extractor data and giving 10 or 3 distinct values based on weights. This classification head gives us a final vector result of class probabilities from which we use the highest value as our predictions. The model is compiled with the Adam optimizer and a learning rate of 0.0001, ready for training.

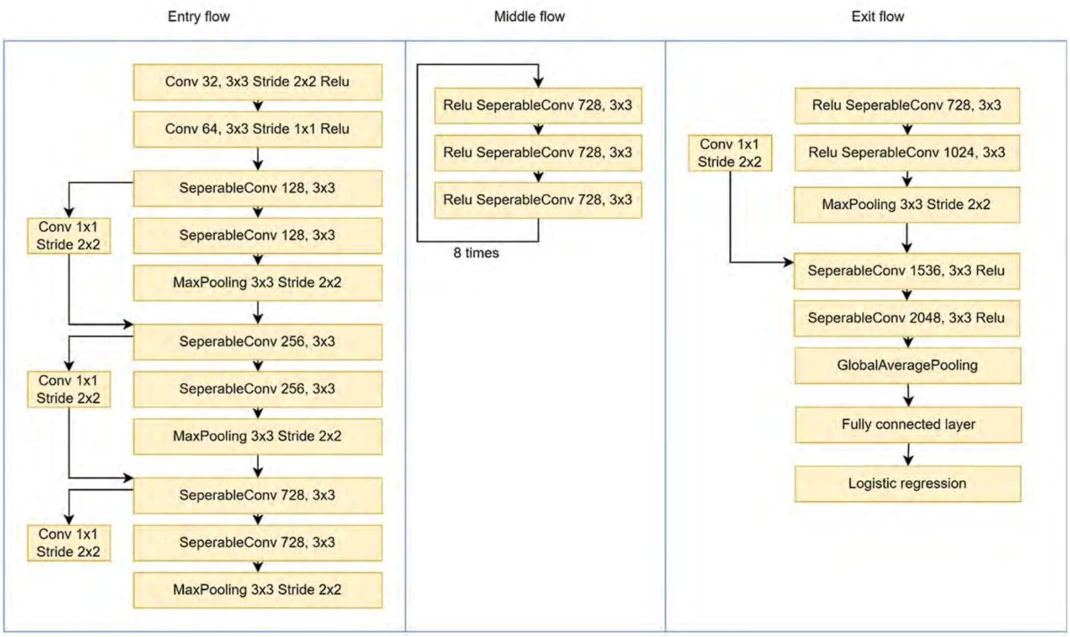


Figure 5.1: Architecture of XceptionNet.

5.1.2 VGG19

The next model we have is based on the VGG19 architecture with transfer learning from a pre-trained model to leverage the advantages as discussed earlier in our Xception based model discussion. The VGG19 model uses 19 layers, 16 convolutional and 3 fully connected, which is a simple architecture using 3x3 convolutional filters stacked together. This makes it effective at capturing details and patterns and it works as a strong feature extractor by using pre-trained weights from ImageNet [35]. The approach here is the same as our Xception based model, the core base model layers are frozen, meaning the weights are preserved during our training, and the final output from the core model is given to our custom classification head. The classification head we have used here is the same as earlier with some minor tweaks we have deemed suitable for this scenario. There is a GlobalAveragePolling2D layer to flatten the output which now flows into a fully connected Dense layer with 512 neurons using a ReLU activation function, introducing non literality and capturing the complexity of the extracted features. This is followed by our Dropout layer with a rate of 0.2 which drops 20% of our neurons to prevent overfitting and finally a Dense layer is used with 10 or 3 units, to match the number of labels, with the softmax activation function to produce our probability distribution for predictions. The model uses the same compilation hyperparameters as our earlier model, Adam optimizer, learning rate of 0.0001 and categorical cross entropy loss function [5].

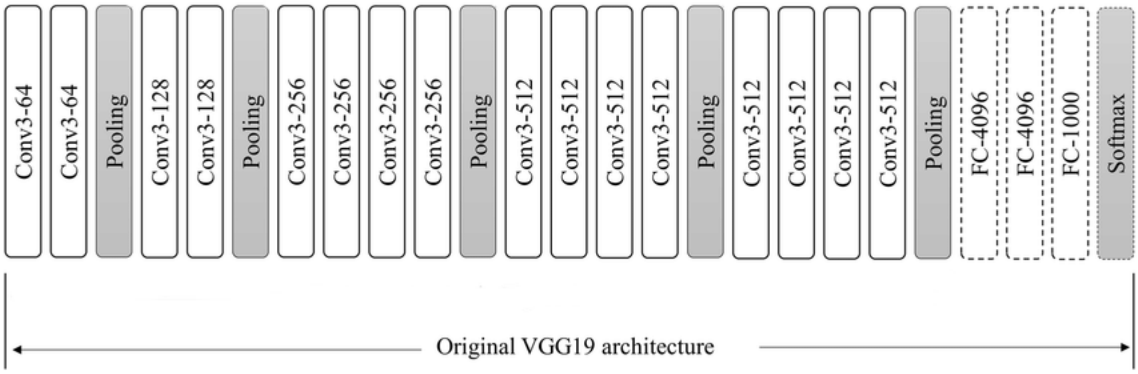


Figure 5.2: Architecture of VGG19.

5.1.3 ResNet50

The last model we have used for testing is ResNet50. A popular convolutional neural network architecture, the 50 in the name stands for 50 layers it uses, it makes use of residual connections also known as skip connections which help mitigate a problem of vanishing gradients for deep models like such especially during back propagation, where gradients become too small - affecting training, and ensures they flow more effectively. The 50 layers consist of 49 convolutional layers and a final fully connected layer, not used in our purpose of transfer learning [10]. Standard transfer learning protocols are used as discussed with earlier models. The custom classification head for this is the same with GlobalAveragePooling2D layer, followed by a BatchNormalization layer and then a Dense layer with 512 units to capture high level features. A Dropout layer has been used here at a higher rate of 0.3 than previously along with another BatchNormalization layer followed by a 0.2 Dropout layer to improve generalization based on observed training performance and finally a 10 or 3 neuron Dense classification layer. No change in our compilation hyperparameters, Adam optimizer, learning rate of 0.0001 and categorical cross entropy loss function.

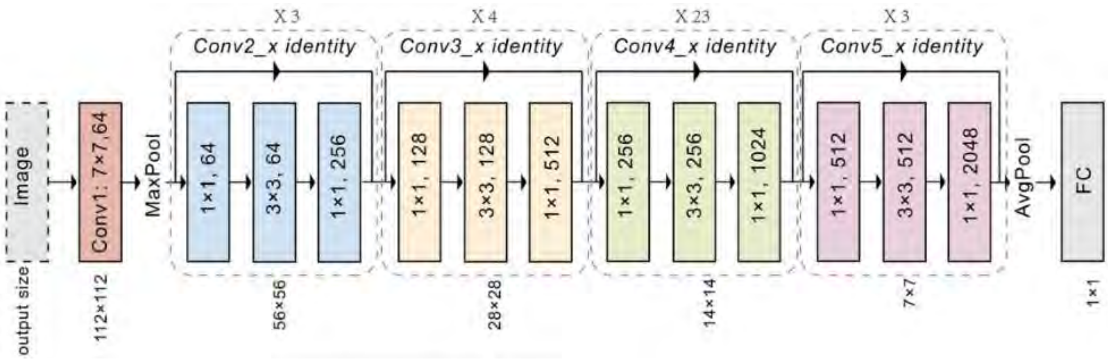


Figure 5.3: Architecture of ResNet50.

5.1.4 Xception + ViT (Hybrid Model)

Through our testing we have seen that the Xception based model performed the best with our datasets, it gave us good metrics which are explored in depth in our results analysis phase. However, we have seen there is room to improve and as such we have implemented a hybrid model combining our feature extractor convolutional neural network, Xception, with a Vision Transformer (ViT) with sequential modelling capabilities. ViTs, created to be an alternative to convolutional networks in vision tasks, particularly NLPs, use a sequence of tokens and attention mechanisms to capture the relationship between pairs [38]. In this model the Xception model outputs a feature map which is transformed into a lower dimension by a Conv2D layer with a 1x1 filter and vit-dimension filters to get a desired shape for the ViT. The feature map is reshaped into a 2D sequence of patch embeddings with batch size, number of patches and vit-dimensions representing a patch of the spatial feature maps. This allows us to use Xception with convolutional inductive bias for low-level feature extraction and then moving to transformers to leverage high level feature extraction capabilities. Positional embedding is also done for the transformer to understand spatial order. The transformer uses a sequence of encoder blocks, namely Multi Head Self Attention which enables capturing of global interactions between tokens, a Feed Forward network to refine the attention layer output and Layer Normalization block to stabilize inputs for attention, prevent activation values drifting too far and help training. The architecture can be seen in diagram below. The transformer output is then fed through a classification head by first passing the patch embedding data through Layer Normalization, it is then sent through GlobalAveragePooling1D to aggregate information from the token data into a single vector for our Dense Layers, first one being 256 units to refine the vector data, then a Dropout of 0.3 followed by our final Dense layer with number of classes to produce our probabilities[16]. Compilation hyperparameters have been kept the same as before: Adam optimizer, learning rate of 0.0001 and categorical cross entropy loss function in line with our multiclass classification problem. A breakdown of our model's layers and workflow can be seen in Figure 5.4.

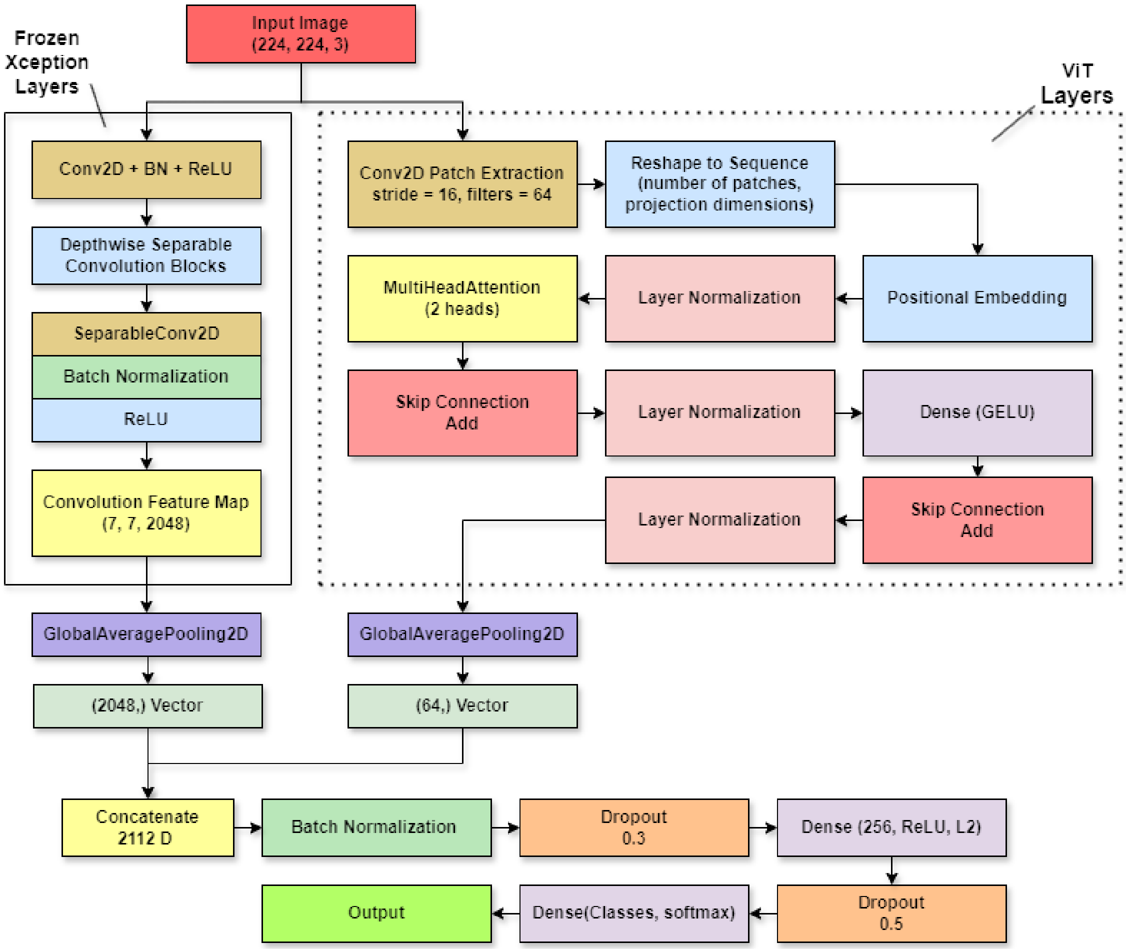


Figure 5.4: Xception + ViT

5.2 Model Result Analysis

5.2.1 Introduction

As discussed we have implemented 3 models, XceptionNet, VGG19 and ResNet50. Based on previous papers and related works consulted in our earlier reports we have come to the conclusion these models perform the best and are the current state of the art based on our current use case and needs. These are fairly robust especially for our limited dataset and via transfer learning we are able to harness their power as we can call upon their experience with general image processing being trained on the imagenet dataset as discussed earlier. Our results have shown amicable validation accuracy scores for the preliminary stage after running them for 30 epochs. Early Stopping is also implemented to stop the training run before overfitting. Each model has been run on each of our datasets for a total of 9 runs. The results for each model and dataset are discussed as follows.

5.2.2 Training Steps

To train our models we have used some standard parameters alongside some additional functions to improve generalization during training. As discussed earlier in the dataset section we have created augmented training sets and validation sets. Using these sets we have implemented the fit() function alongside implementing class weights to address class imbalances. In addition to these standard parameters we have used two special strategies or callbacks, namely early stopping and learning rate reduction with the help of the functions EarlyStopping and ReduceLROnPlateau. EarlyStopping is useful to stop training if there is no further improvement, observing validation loss to determine this, after a number of epochs which in our case the patience is set at 5 meaning 5 continuous instances of no improvement will result in the training halting. This helps save resources wasted to run additional epochs with no discernable improvement. ReduceLROnPlateau comes in to mitigate this above scenario by doing on the fly learning rate adjustments, it observes for plateauing and lack of improvement of validation loss. If 3 consecutive epochs, in our case, see no improvement then it reduces the learning rate by a factor of 0.2 and the minimum learning rate can be 0.00001. This helps combat overfitting and ensure training effectiveness. Finally the model is run for 30 epochs unless the EarlyStopping callback is made in case of no further possible improvement.

5.2.3 Results Collection

From our training phase we have collected the following metrics for training and validation:

- Accuracy
- Loss
- Precision
- Recall

After which during our validation phase on unseen data from which we have collected confusion matrix and classification report to understand how our model performs on unseen data. It provides us with the F-1 Score.

$$\text{Loss} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2.4 Xception Results

The following are the results from our Xception transfer learning approach, as Figure 5.5 and 5.6 show the training went smoothly and shows no overfitting or underfitting. The final validation results are shown in the Table 5.1.

Dataset	Accuracy	Loss	Precision	Recall
Rice	0.916	0.246	0.937	0.897
Potato	0.906	0.204	0.921	0.899
Wheat	0.981	0.057	0.982	0.981

Table 5.1: Training Data for Xception Model.

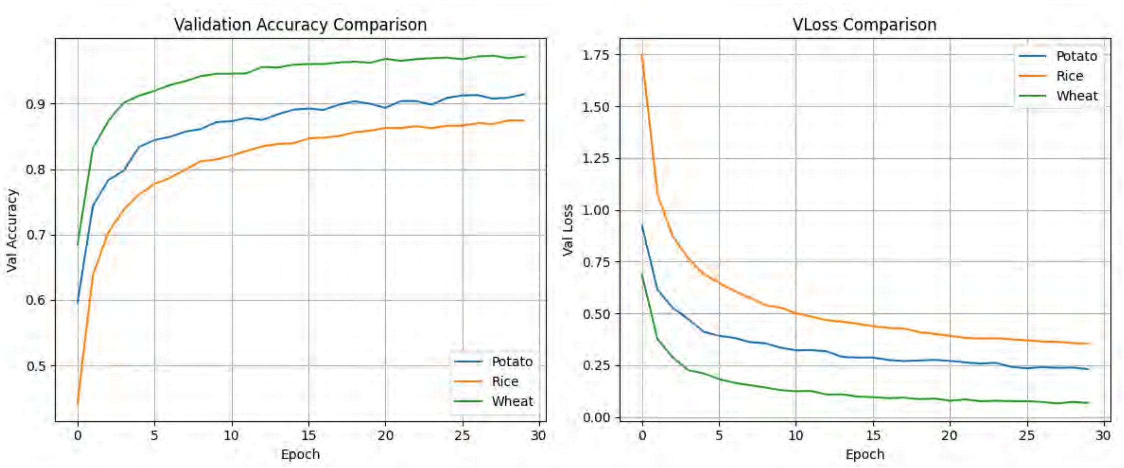


Figure 5.5: Training Validation Accuracy and Loss Graphs for Xception.

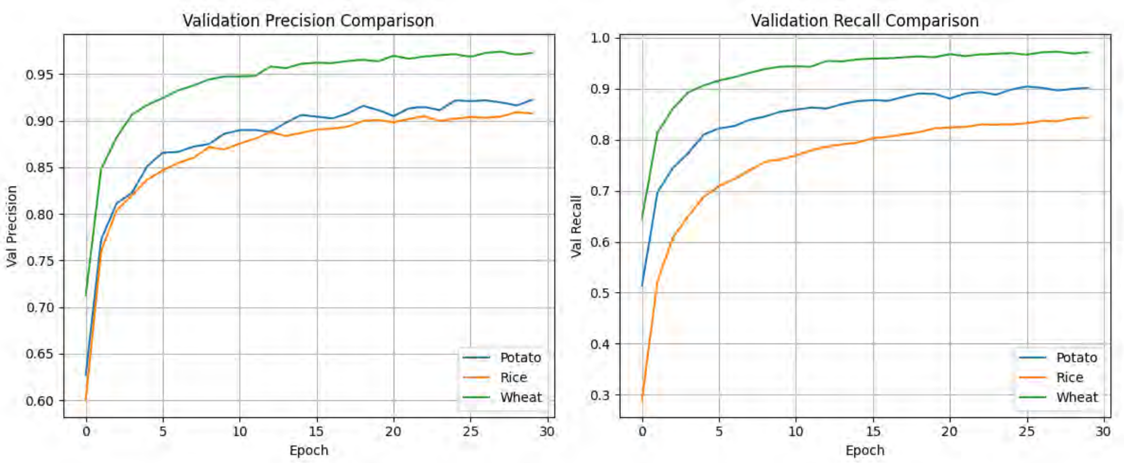


Figure 5.6: Training Validation Precision and Recall Graphs for Xception.

Xception transfer learning approach results show good generalization to our data, especially wheat. The reason it did so well for wheat we assume is the use of more intense augmentation techniques and overall more well rounded dataset used in our training. The graphs showing our final precision and recall values show training had been done correctly giving us a valid result. Afterwards we have conducted classification tests to see how well our model performs on unseen data, the reports for which are in Tables 5.2, 5.3 and 5.4. The confusion matrices are shown in Figures 5.7, 5.8 and 5.9.The model performed well on our unseen data as well, reinforcing our training results.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Bacterial Leaf Blight	0.99	0.98	0.98	140
Brown Spot	0.95	0.85	0.90	148
Healthy	0.86	0.95	0.90	158
Leaf Blast	0.86	0.80	0.83	181
Leaf Scald	0.98	0.92	0.95	167
Narrow Brown Spot	0.88	0.88	0.88	143
Neck Blast	1.00	1.00	1.00	108
Rice Hispa	0.92	0.89	0.91	141
Sheath Blight	0.86	0.94	0.90	159
Tungro	0.98	0.99	0.99	174
Accuracy			**0.92**	**1509**
Macro Avg	0.92	0.92	0.92	1509
Weighted Avg	0.92	0.92	0.92	1509

Table 5.2: Classification Report for Rice Dataset on Xception.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Early Blight	0.97	0.93	0.95	162
Healthy	0.90	0.91	0.91	102
Late Blight	0.90	0.94	0.92	141
Accuracy			**0.93**	**405**
Macro Avg	0.93	0.93	0.93	405
Weighted Avg	0.93	0.93	0.93	405

Table 5.3: Classification Report for Potato Dataset on Xception.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Healthy	0.97	1.00	0.98	226
Septoria	1.00	1.00	1.00	216
Stripe Rust	1.00	0.98	0.99	401
Accuracy			**0.99**	**843**
Macro Avg	0.99	0.99	0.99	843
Weighted Avg	0.99	0.99	0.99	843

Table 5.4: Classification Report for Wheat Dataset on Xception.

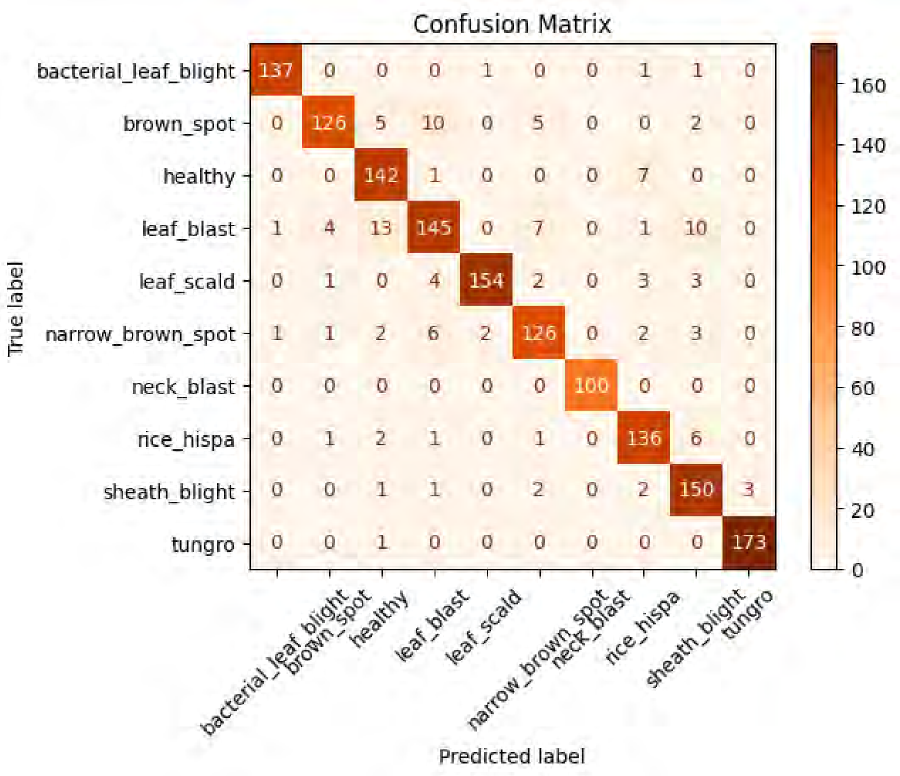


Figure 5.7: Confusion Matrix for Rice Dataset on Xception.

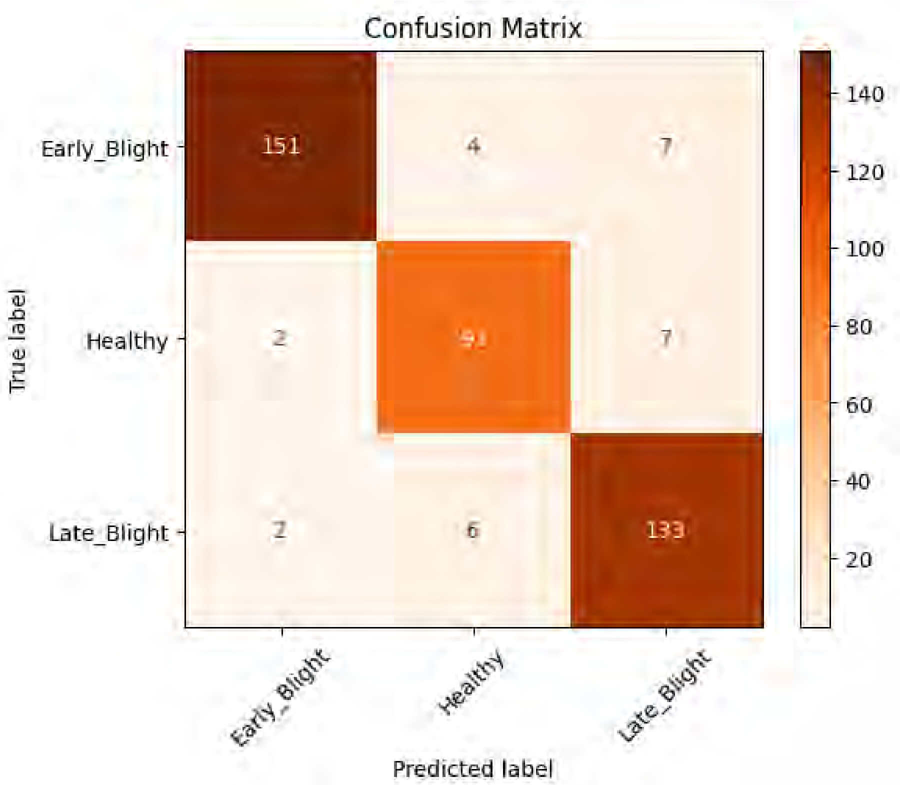


Figure 5.8: Confusion Matrix for Potato Dataset on Xception.

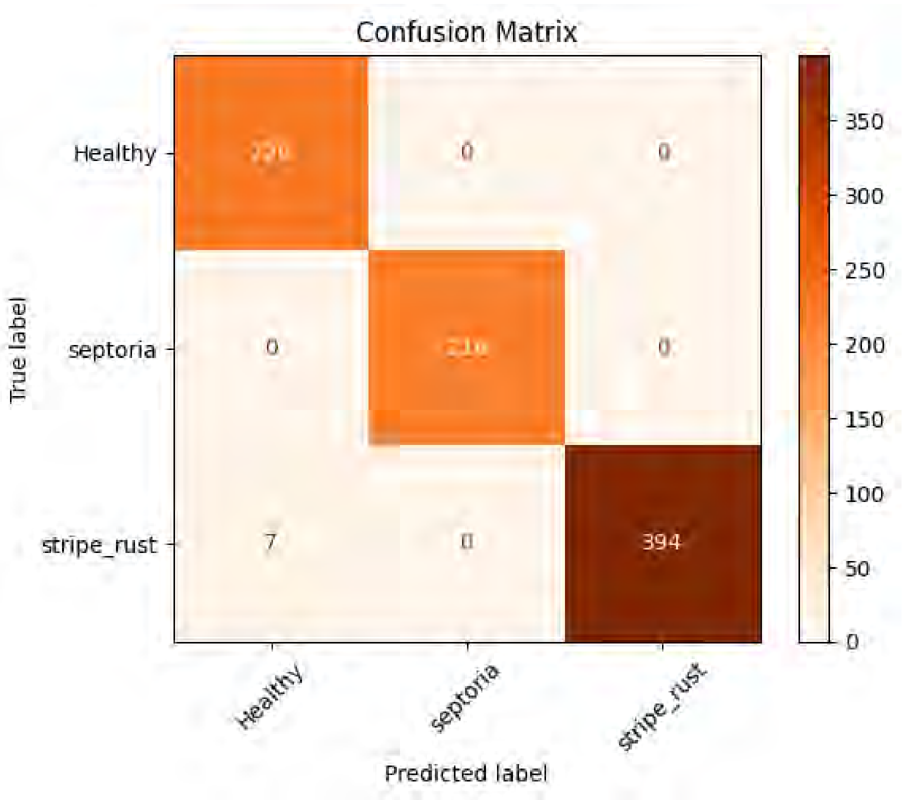


Figure 5.9: Confusion Matrix for Wheat Dataset on Xception.

5.2.5 VGG19 Results

The following are the results from our VGG19 transfer learning approach, as Figure 5.10 and Figure 5.11 show the training went smoothly and shows no overfitting or underfitting. The final validation results are shown in the Table 5.5.

Dataset	Accuracy	Loss	Precision	Recall
Rice	0.801	0.570	0.886	0.715
Potato	0.819	0.489	0.870	0.742
Wheat	0.978	0.069	0.978	0.978

Table 5.5: Training Data for VGG19 Model.

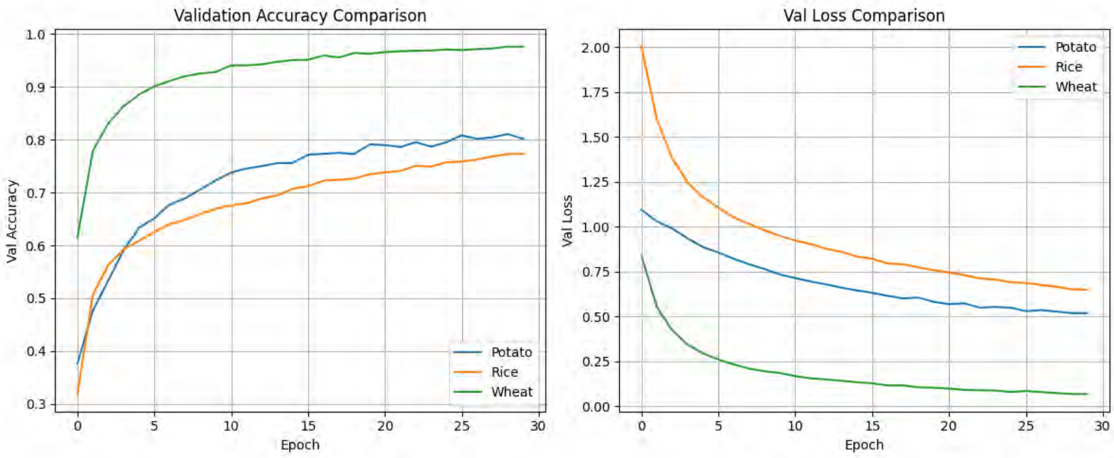


Figure 5.10: Training Validation Accuracy and Loss Graphs for VGG19.

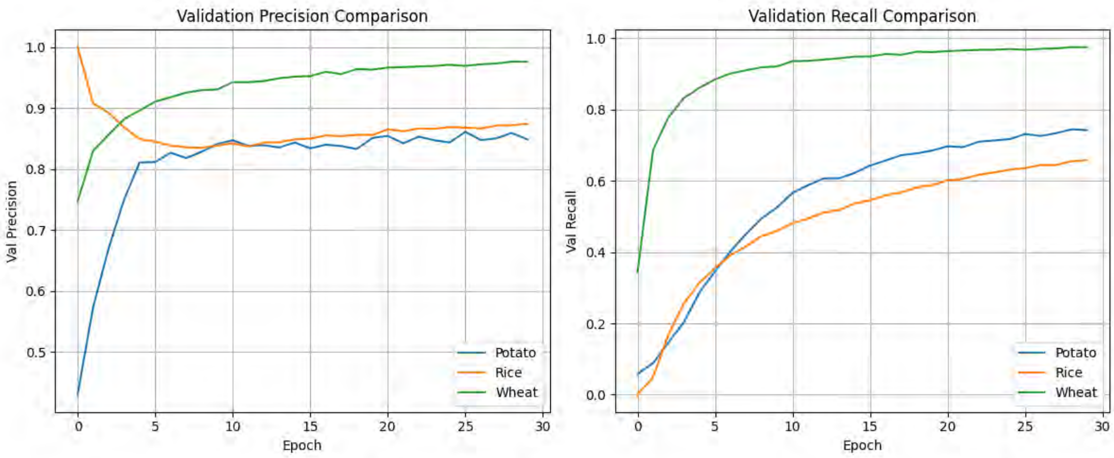


Figure 5.11: Training Validation Precision and Recall Graphs for VGG19.

VGG19 transfer learning approach results show good generalization to our data, however not on par with Xception. It got close but maybe the model being too simple did not allow it capture our required details properly. The reports for classification test on unseen data are in Tables 5.6, 5.7, and 5.8. The confusion matrices are shown in Figures 5.12, 5.13, and 5.14,.The model performed well on our unseen data as well, reinforcing our training results.

Class	Precision	Recall	F1 Score	Support
Bacterial Leaf Blight	0.81	0.92	0.86	140
Brown Spot	0.76	0.66	0.71	148
Healthy	0.66	0.85	0.74	150
Leaf Blast	0.78	0.50	0.61	181
Leaf Scald	0.88	0.82	0.85	167
Narrow Brown Spot	0.74	0.78	0.76	143
Neck Blast	0.98	1.00	0.99	109
Rice Hispa	0.72	0.86	0.79	147
Sheath Blight	0.78	0.81	0.80	159
Tungro	0.95	0.91	0.93	174
Accuracy			**0.80**	1509
Macro Average	0.81	0.81	0.80	1509
Weighted Average	0.80	0.80	0.80	1509

Table 5.6: Classification Report for Rice Dataset on VGG19.

Class	Precision	Recall	F1 Score	Support
Early Blight	0.87	0.85	0.86	162
Healthy	0.86	0.81	0.84	102
Late Blight	0.79	0.84	0.82	141
Accuracy			**0.84**	405
Macro Average	0.84	0.84	0.84	405
Weighted Average	0.84	0.84	0.84	405

Table 5.7: Classification Report for Potato Dataset on VGG19.

Class	Precision	Recall	F1 Score	Support
Healthy	0.97	1.00	0.98	226
Septoria	0.99	1.00	1.00	216
Stripe Rust	1.00	0.98	0.99	401
Accuracy			**0.99**	843
Macro Average	0.99	0.99	0.99	843
Weighted Average	0.99	0.99	0.99	843

Table 5.8: Classification Report for Wheat Dataset on VGG19.

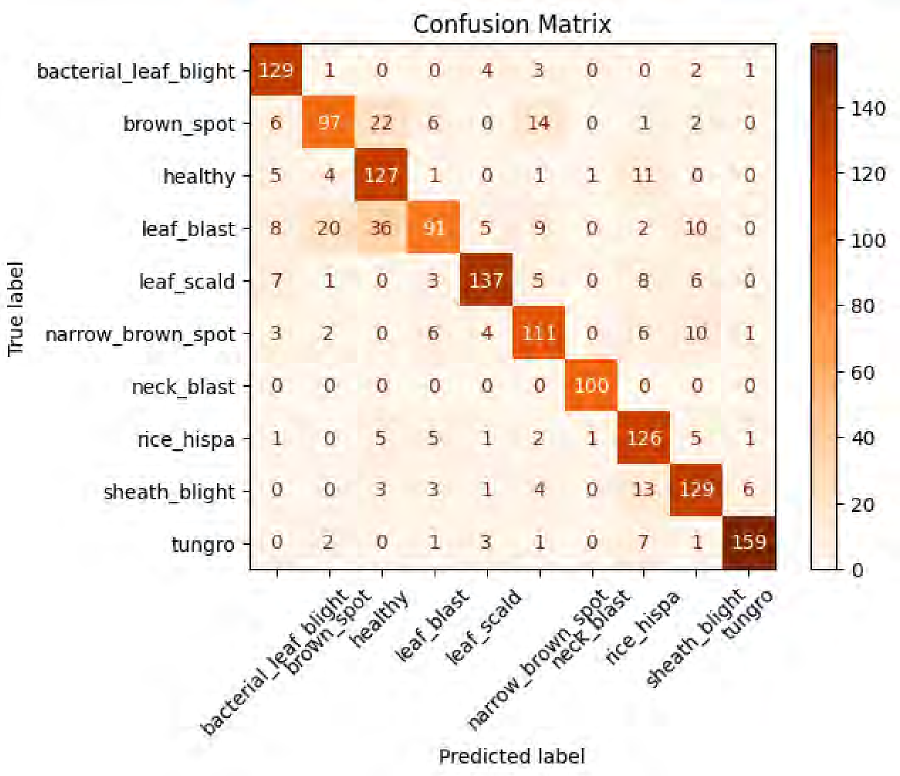


Figure 5.12: Confusion Matrix for Rice Dataset on VGG19.

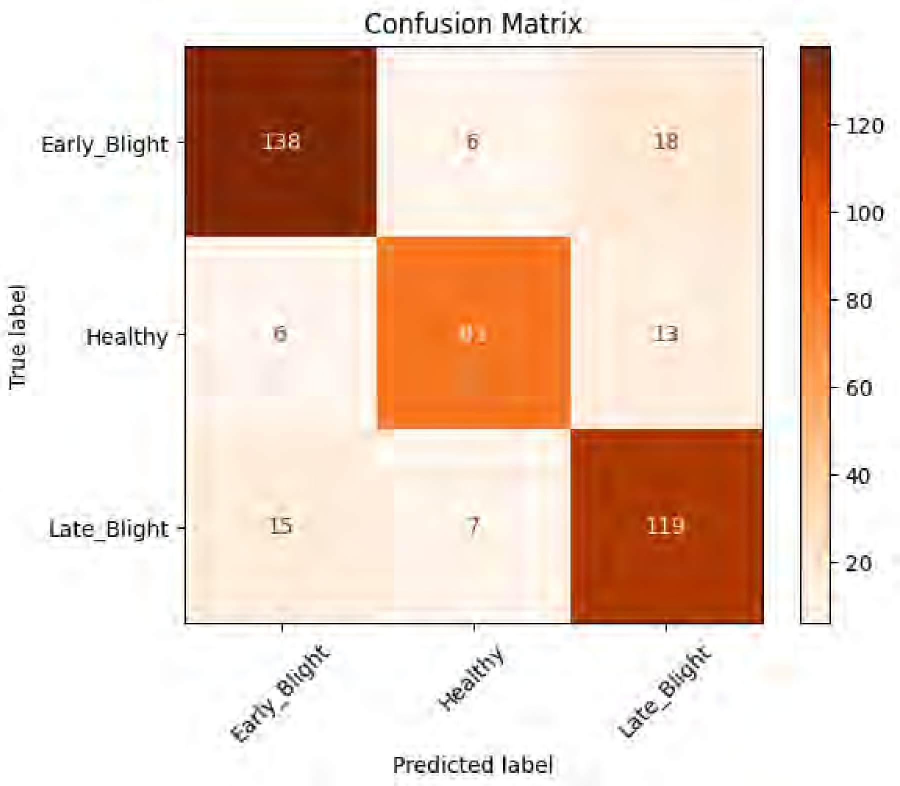


Figure 5.13: Confusion Matrix for Potato Dataset on VGG19.

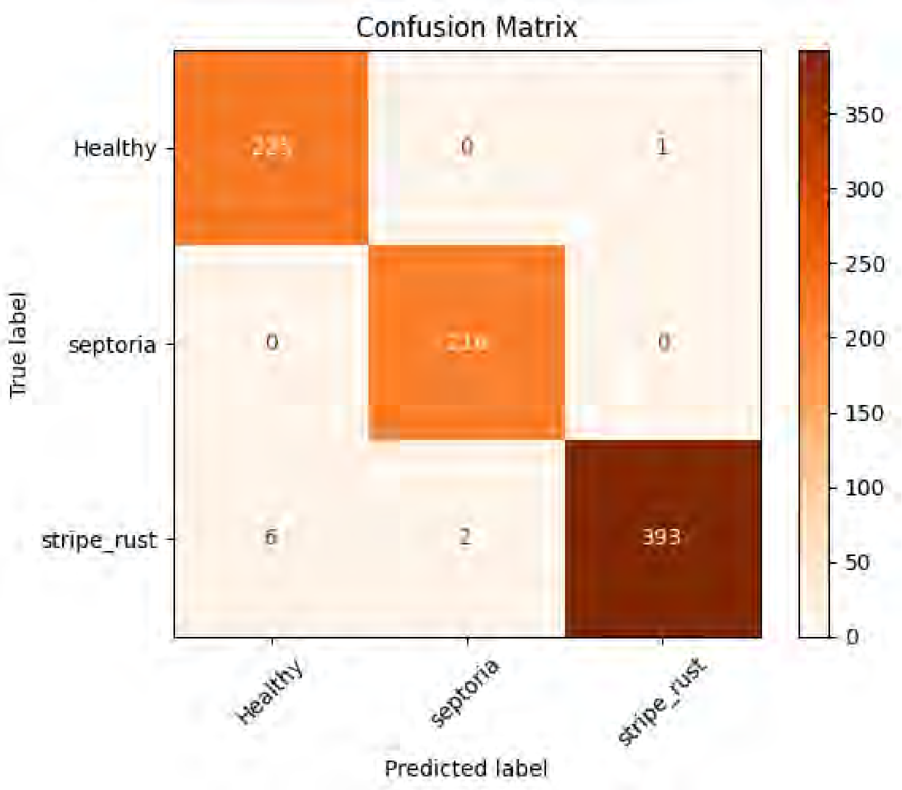


Figure 5.14: Confusion Matrix for Wheat Dataset on VGG19.

5.2.6 ResNet50 Results

The following are the results from our ResNet50 transfer learning approach, as Figure 5.15 and Figure 5.15 show the training went smoothly and shows no overfitting or underfitting. The final validation results are shown in the Table 5.9.

Dataset	Accuracy	Loss	Precision	Recall
Rice	0.797	0.608	0.862	0.728
Potato	0.771	0.609	0.800	0.723
Wheat	0.973	0.066	0.973	0.972

Table 5.9: Training Data for ResNet50 Model.

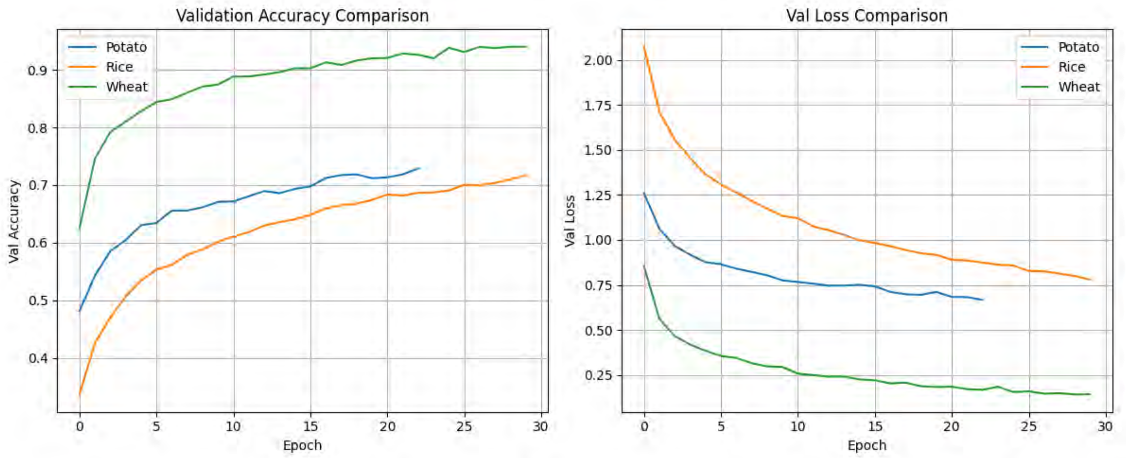


Figure 5.15: Training Validation Accuracy and Loss Graphs for ResNet50.

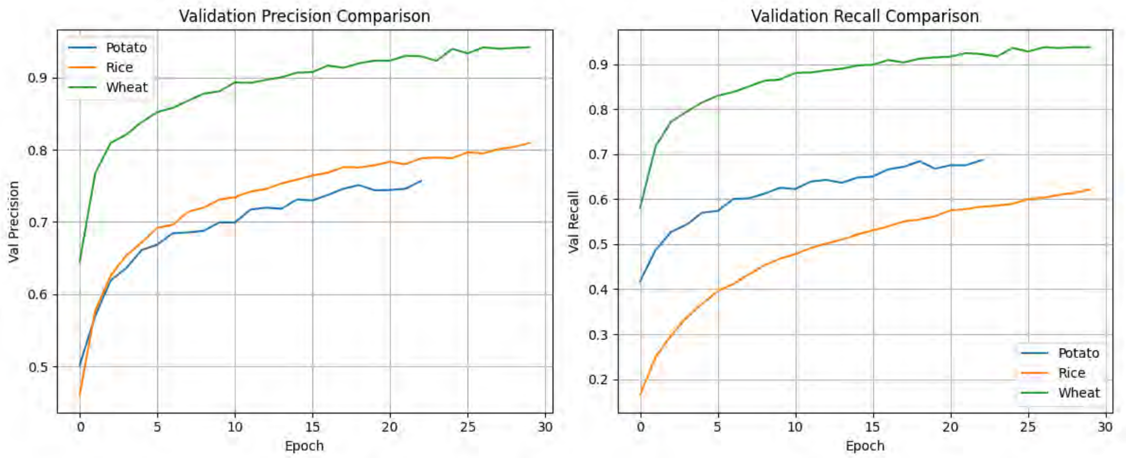


Figure 5.16: Training Validation Precision and Recall Graphs for ResNet50.

ResNet50 did not perform as well as our other two models. The reason for this could be the more complex nature of the problem with fine grained patterns present in our disease data which the smooths over. It is not as deep as our other two tested models. The classification report for ResNet50 is shown in Tables 5.10, 5.11 and 5.12. The confusion matrices are shown in Figures 5.17, 5.18 and 5.19.The model performed well in accordance to its training data on the unseen data classification tasks.

Class	Precision	Recall	F1 Score	Support
Bacterial Leaf Blight	0.88	0.95	0.91	140
Brown Spot	0.80	0.69	0.74	148
Healthy	0.65	0.81	0.72	156
Leaf Blast	0.76	0.47	0.58	181
Leaf Scald	0.83	0.80	0.81	143
Narrow Brown Spot	0.64	0.81	0.71	143
Neck Blast	0.96	0.99	0.98	104
Rice Hispa	0.80	0.35	0.49	108
Sheath Blight	0.74	0.81	0.77	182
Tungro	0.97	0.88	0.92	174
Accuracy			**0.79**	1509
Macro Average	0.80	0.81	0.80	1509
Weighted Average	0.80	0.79	0.79	1509

Table 5.10: Classification Report for Rice Dataset on ResNet50.

Class	Precision	Recall	F1 Score	Support
Early Blight	0.79	0.78	0.79	162
Healthy	0.79	0.74	0.76	102
Late Blight	0.72	0.77	0.74	141
Accuracy			**0.76**	405
Macro Average	0.77	0.76	0.76	405
Weighted Average	0.76	0.76	0.76	405

Table 5.11: Classification Report for Potato Dataset on ResNet50.

Class	Precision	Recall	F1 Score	Support
Healthy	0.97	0.99	0.98	226
Septoria	0.99	1.00	1.00	216
Stripe Rust	0.99	0.98	0.99	401
Accuracy			**0.99**	843
Macro Average	0.99	0.99	0.99	843
Weighted Average	0.99	0.99	0.99	843

Table 5.12: Classification Report for Wheat Dataset on ResNet50.

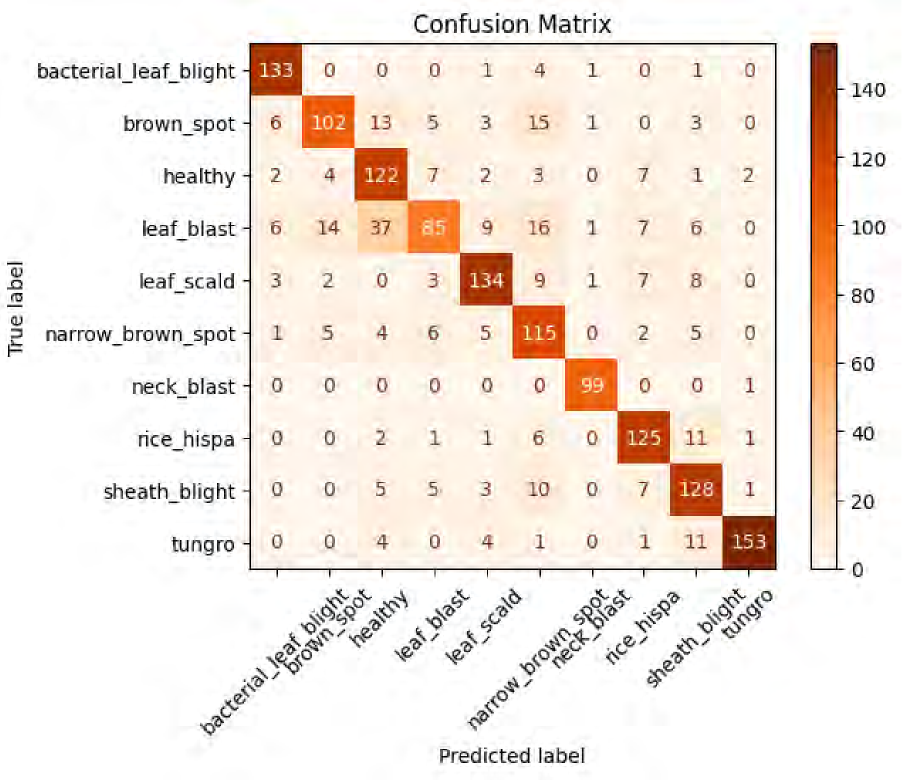


Figure 5.17: Confusion Matrix for Rice Dataset on ResNet50.

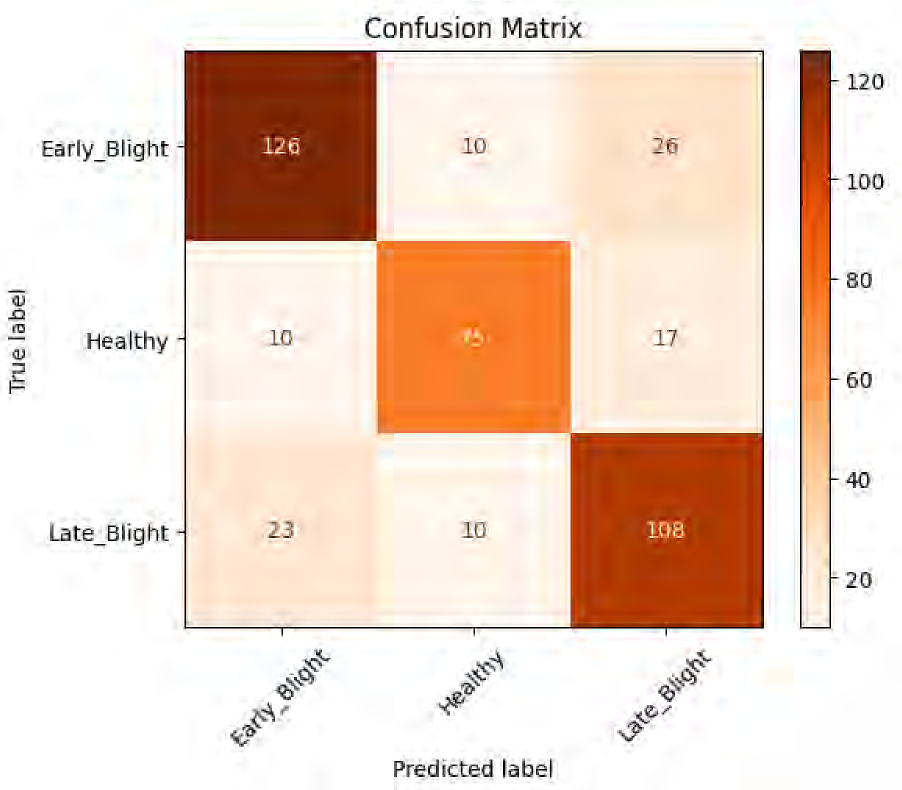


Figure 5.18: Confusion Matrix for Potato Dataset on ResNet50.

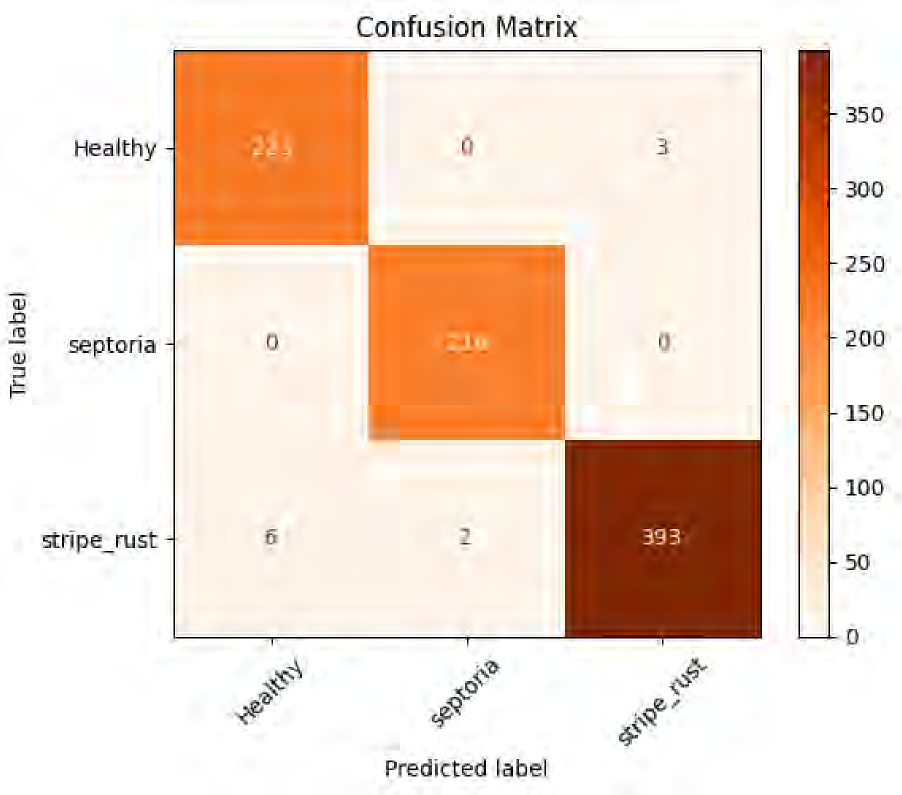


Figure 5.19: Confusion Matrix for Wheat Dataset on ResNet50.

5.2.7 Xception + ViT (Hybrid Model) Results

After testing our three models we have decided to improve our Xception model as it had the best results by adding a transformer and creating a hybrid model. This approach has allowed us to achieve better results as shown in the following Table 5.13. The training data is also promising as shown in Figures 5.20 and 5.21.

Dataset	**Accuracy**	**Loss**	**Precision**	**Recall**
Rice	0.957	0.167	0.965	0.949
Potato	0.978	0.110	0.980	0.973
Wheat	0.998	0.053	0.998	0.998

Table 5.13: Training Data for Hybrid Model.

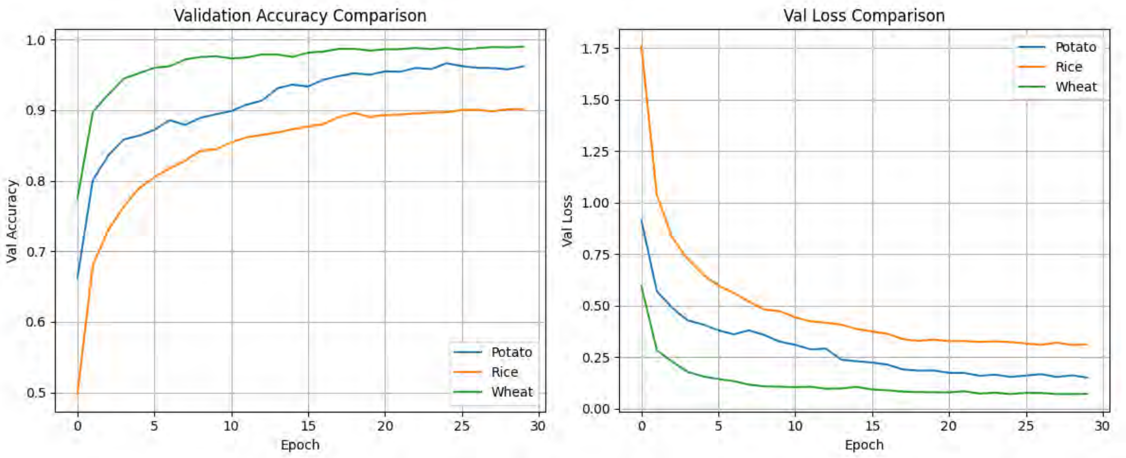


Figure 5.20: Training Validation Accuracy and Loss Graphs for Hybrid Model.

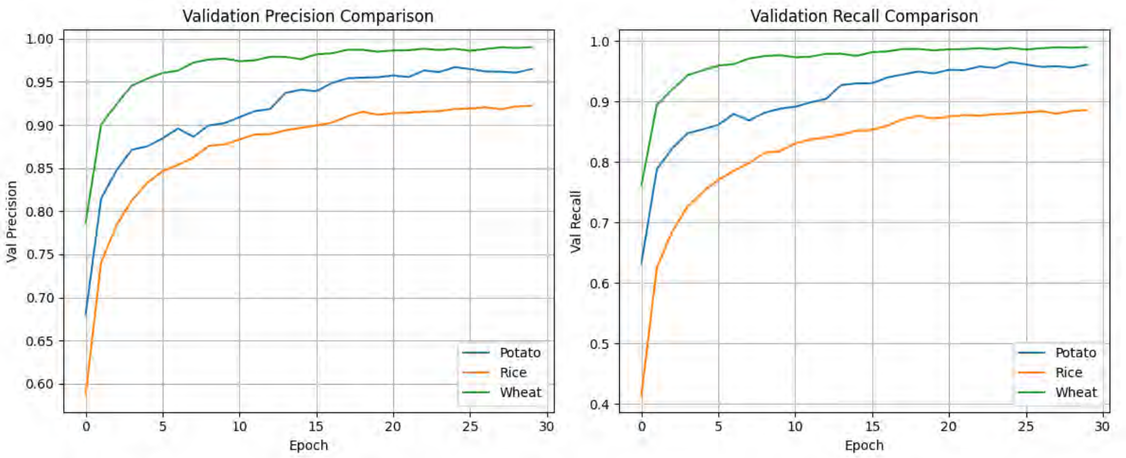


Figure 5.21: Training Validation Precision and Recall Graphs for Hybrid Model.

The final hybrid model approach we have here, combines Xception and ViT as discussed earlier. Levraging the low level feature extraction capability of CNNs with high level feature exctraction capability of ViTs. The results during training are better than our tested models shows this is viable. Classification reports are shown in Tables 5.14, 5.15 and 5.16. The confusion matrices are shown in Figures 5.22, 5.23 and 5.24. The results further show the capability of our model even at unseen data classification.

Class	Precision	Recall	F1 Score	Support
Bacterial Leaf Blight	1.00	0.99	1.00	140
Brown Spot	0.95	0.96	0.95	148
Healthy	0.95	0.99	0.97	140
Leaf Blast	0.97	0.91	0.94	181
Leaf Scald	0.99	0.97	0.98	167
Narrow Brown Spot	0.93	0.97	0.95	143
Neck Blast	1.00	1.00	1.00	103
Rice Hispa	0.99	0.98	0.98	104
Sheath Blight	0.96	0.97	0.97	159
Tungro	0.99	1.00	1.00	174
Accuracy			**0.97**	1509
Macro Average	0.97	0.97	0.97	1509
Weighted Average	0.97	0.97	0.97	1509

Table 5.14: Classification Report for Rice Dataset on Hybrid Model.

Class	Precision	Recall	F1 Score	Support
Early Blight	0.99	0.95	0.97	162
Healthy	0.94	0.97	0.96	102
Late Blight	0.97	0.99	0.98	141
Accuracy			**0.97**	405
Macro Average	0.97	0.97	0.97	405
Weighted Average	0.97	0.97	0.97	405

Table 5.15: Classification Report for Potato Dataset on Hybrid Model.

Class	Precision	Recall	F1 Score	Support
Healthy	1.00	1.00	1.00	226
Septoria	1.00	1.00	1.00	216
Stripe Rust	1.00	1.00	1.00	401
Accuracy			**1.00**	843
Macro Average	1.00	1.00	1.00	843
Weighted Average	1.00	1.00	1.00	843

Table 5.16: Classification Report for Wheat Dataset on Hybrid Model.

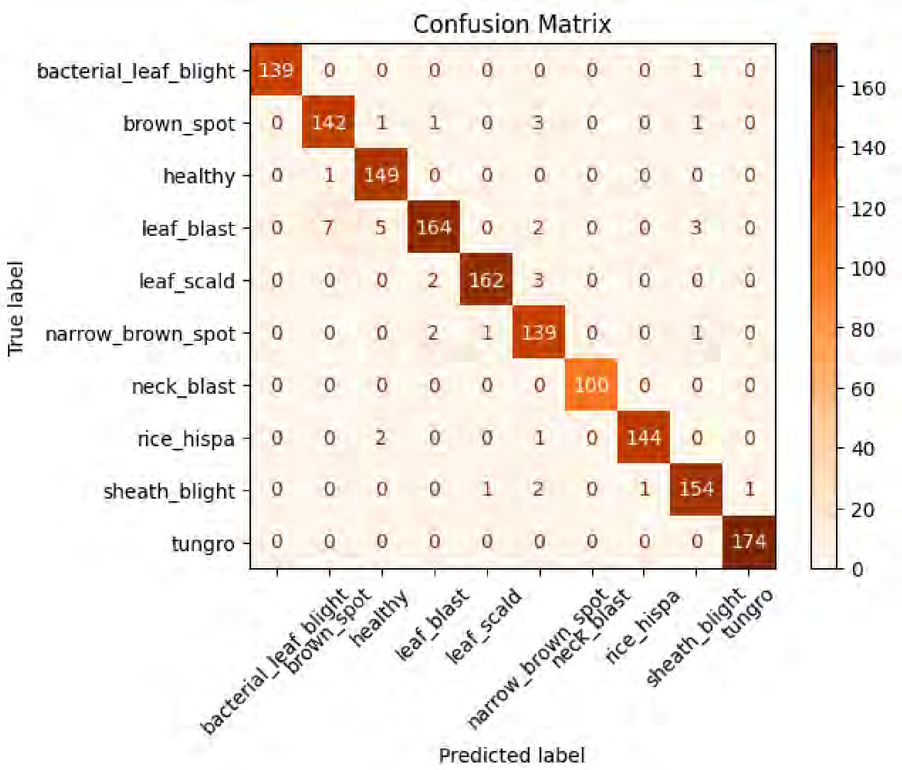


Figure 5.22: Confusion Matrix for Rice Dataset on Hybrid Model.

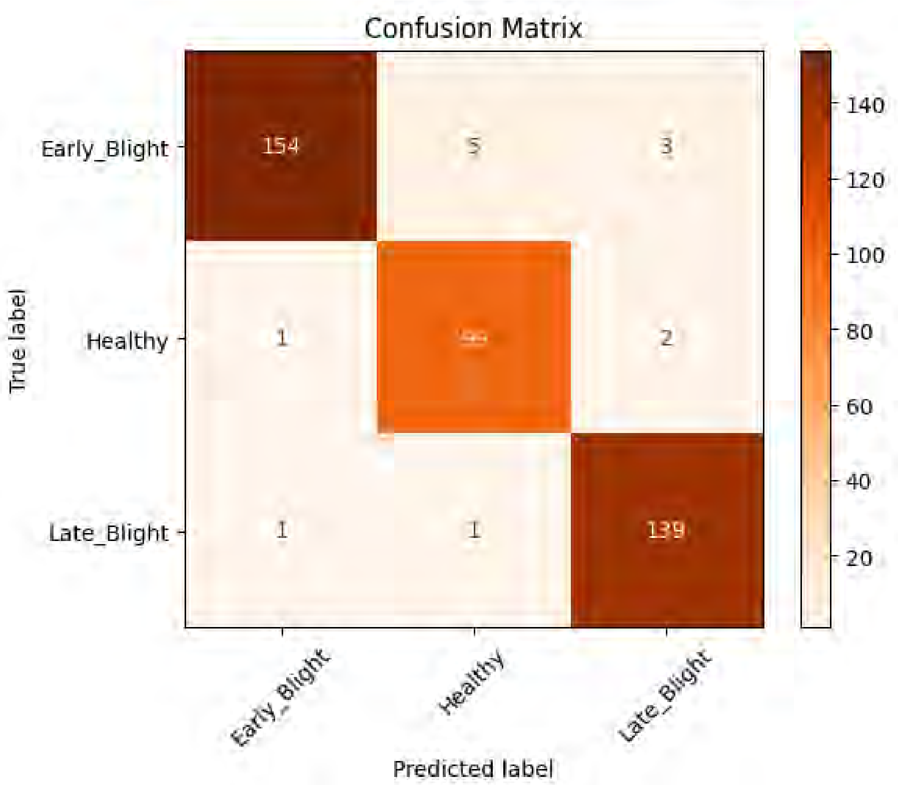


Figure 5.23: Confusion Matrix for Potato Dataset on Hybrid Model.

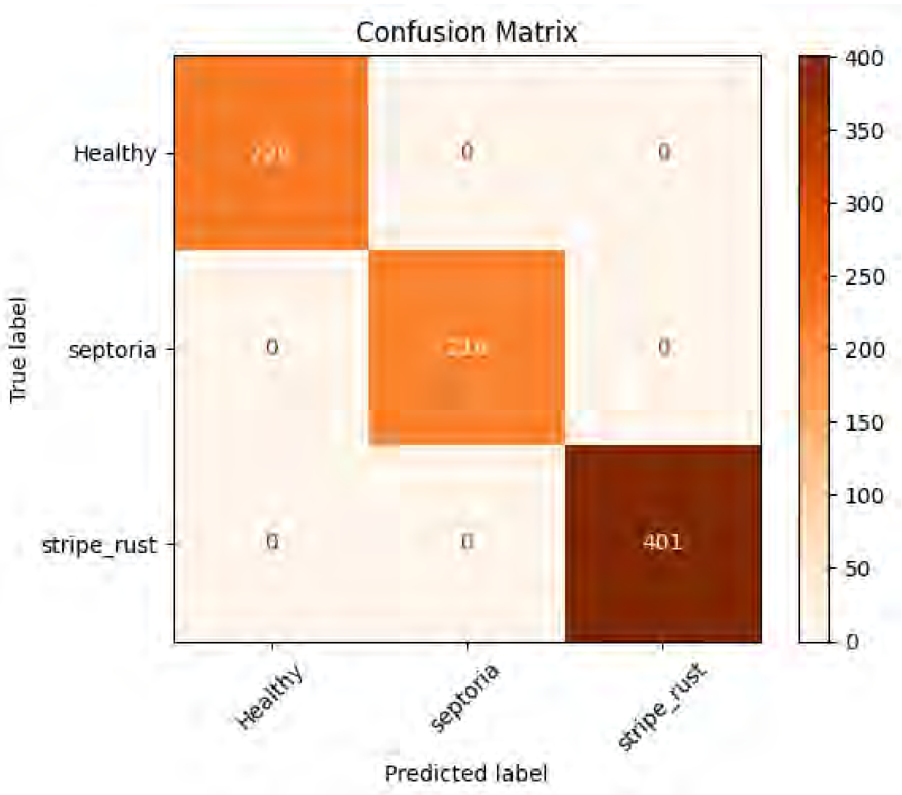


Figure 5.24: Confusion Matrix for Wheat Dataset on Hybrid Model.

5.2.8 Comparison With Other Approaches

Lastly, we have compared our model to some other approaches by papers we have reviewed. In conclusion our model performs similarly, if not better compared to the other approaches. The benefit our model has compared to the other approaches is that it uses data that is far newer with more noise and artifacting reflective of real world conditions. This allows our model to be better suited to field applications than others which have been trained on data from simpler and cleaner datasets such as PlantVillage dataset.

Paper	**Dataset**	**Acc.**	**F1**	**Comment**
ResNet-50 + SAM [**22**]	PlantVillage-Potato	96.06%	0.98	Using dataset with curated clean images with minimal noise and artifacting.
	PlantVillage-Wheat	94.08%	0.95	
ResNet-50 [**21**]	PlantVillage-Potato	97.8%	-	Using dataset with curated clean images with minimal noise and artifacting.
EfficientNetV2L [**20**]	PlantVillage-Wheat	96.5%	-	Using dataset with curated clean images, no noise present.
Hybrid (Xception+ViT) (Our Approach)	Wheat Leaf Dataset	99.8%	1.00	Hybrid model shows robustness to noise and illumination changes in real world data.
	Rice Leaf Dataset	97.8%	0.97	
	Potato Leaf Dataset	95.7%	0.97	

Table 5.17: Comparison of published models and datasets with performance and notes.

5.2.9 Explainable AI and GradCAM

To help us understand how our proposed model makes predictions and see images from its point of view we can implement an explainable AI strategy. The explainable AI strategy we have used for this purpose is Grad-CAM. The way Grad-CAM operates is it highlights which part of the image influenced the decision of our model to classify it one way. The way this works is it gets the class prediction scores and the feature map of the last convolutional layer, using which it computes the gradient. This is then used to obtain the spatial importance weights, multiplied with the feature map and corresponding weights and then a ReLU function is applied to keep positive contributions or supporting areas for the decision made. A heatmap is generated which is overlaid on the classified image to show which area contributed to the decision [34].

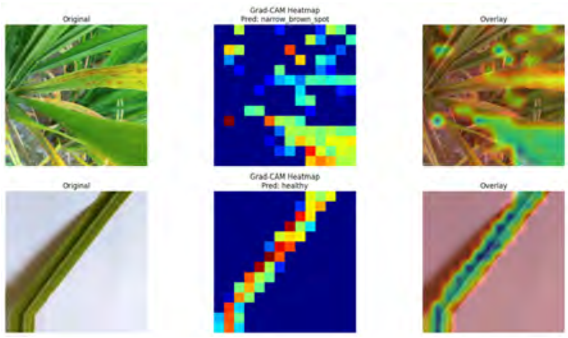


Figure 5.25: GradCAM

The above figure shows an example where our GradCAM heatmap is determined to be an important feature to consider in the decision-making process.

Chapter 6

Conclusion, Limitation and Future Work

6.1 Conclusion

Therefore, deploying AI methodologies for crop disease identification and control marks the shift within an essential improvement in the agricultural division of Bangladesh. By applying such approaches like image processing and machine learning models and algorithms in the given manner, my methodology mainly tries to improve the ability of disease detection not only in rice, wheat, and potatoes but also this methodology is tries to develop a model that could predict in any condition. The possible implication of this approach is to diagnose diseases in crops early and correctly to make necessary actions. This could go a long way in enhancing the economic status of the farmers and the country and reduce the cases of crop damage. The following is an indication of how we have ensured that course towards a resilience sustainable agricultural ecosystem; The used AI technologies. We believe that such an approach would definitely bring a drastic shift in peoples perception towards farming and has the potential to transform the crop management market.

6.2 Challenges and Limitations

6.2.1 Insufficient Dataset Diversity

Although the datasets are sufficient for preliminary testing, they are not diverse enough for various geographical locations, environmental circumstances, or soil types. Adding more variety of samples would greatly increase the accuracy and reliability of the model, as crop diseases might present differently depending on the location, climate, or soil type. A primary dataset consisting of more field samples suited to our conditions would be beneficial.

6.2.2 Difficulty in Class Differentiation

The model sometimes misclassified certain crop diseases because of their similar visual patterns. More variation is needed in the training data by experimenting with more augmentation techniques as we have seen in the wheat dataset it benefited from greater number of augmentations.

6.2.3 Hardware Limitations

The models could be trained better if we had access to unrestricted hardware time, which would allow us to train our model for greater than 30 epochs as each training pass took a great amount of time especially for the larger dataset for Rice. Dataset and data loading adjustments could also be made to help with this issue.

6.2.4 Faulty GradCAM Implementation

Our GradCAM implementation did not give us the expected results from our hybrid model. There may be issues with our implementation that would require more exploration in order to rectify it.

6.3 Future Work

6.3.1 Smartphone Application Integration

Creating a mobile application that is lightweight is one possible area for future development. An application like this would enable farmers to rapidly use the cameras on their phones to scan leaves and obtain instant disease detection, increasing the system's accessibility in low-resource and rural locations.

6.3.2 Detection of Multiple Diseases

Currently, the system identifies a single disease per image. Future enhancements could focus on enabling multi-label classification to detect more than one disease affecting the same leaf, which is common in real-world farming conditions. Training the model on larger varied dataset of more crops can be explored.

6.3.3 Primary Dataset and More Variables

Creation of a primary dataset of not only leaf images but also weather and other control variables for diseases would help us create a model better able to identify diseases.

Bibliography

[1] Olusola Oluwakemi Abayomi-Alli et al. "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning". In: *Expert Systems* 38.7 (2021), e12746.

[2] Husnul Ajra et al. "Disease detection of plant leaf using image processing and cnn with preventive measures". In: *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*. IEEE. 2020, pp. 1–6.

[3] Marko Arsenovic et al. "Solving current limitations of deep learning based approaches for plant disease detection". In: *Symmetry* 11.7 (2019), p. 939.

[4] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 1251–1258. DOI: 10.1109/CVPR.2017.195.

[5] François Chollet and contributors. *Keras Documentation: Layers API — GlobalAveragePooling2D, Dropout, Dense*. https://keras.io/api/layers/. Accessed: 2025-06-18. 2024.

[6] Halil Durmuş, Ece Olcay Güneş, and Mürvet Kırcı. "Disease detection on the leaves of the tomato plants by using deep learning". In: *2017 6th International conference on agro-geoinformatics*. IEEE. 2017, pp. 1–5.

[7] Ahmed Farouk. *Wheat Leaf*. https://www.kaggle.com/datasets/ahmedfaroukksiu/wheat-leaf. Accessed: 2025-06-17. 2022.

[8] Konstantinos P Ferentinos. "Deep learning models for plant disease detection and diagnosis". In: *Computers and electronics in agriculture* 145 (2018), pp. 311–318.

[9] G Geetharamani and Arun Pandian. "Identification of plant leaf diseases using a nine-layer deep convolutional neural network". In: *Computers & Electrical Engineering* 76 (2019), pp. 323–338.

[10] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[11] Khalid M Hosny et al. "Multi-Class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern". In: *IEEE Access* (2023).

[12] Aminul Islam. "Study on major diseases of major crops in Bangladesh". In: (2019).

[13] Md Ashiqul Islam et al. “An automated convolutional neural network based approach for paddy leaf disease detection”. In: *International Journal of Advanced Computer Science and Applications* 12.1 (2021).

[14] Md Tariqul Islam. “Plant disease detection using CNN model and image processing”. In: *Int. J. Eng. Res. Technol* 9.10 (2020), pp. 291–297.

[15] KC Kamal et al. “Depthwise separable convolution architectures for plant disease classification”. In: *Computers and electronics in agriculture* 165 (2019), p. 104948.

[16] Salman Khan et al. *Transformers in Vision: A Survey.* https://arxiv.org/abs/2101.01169. Discusses hybrid CNN-Transformer architectures including convolutional front-ends (e.g., Xception) with Transformer backbones (e.g., ViT). Accessed: 2025-06-18. 2022.

[17] Sergejs Kodors et al. “Apple scab detection using CNN and Transfer Learning”. In: (2021).

[18] Vinay Kukreja, Vikrant Sharma, and Rishabh Sharma. “Elevating Crop Protection: Vision Transformer Models for Brassica Black Rot Detection”. In: *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE).* IEEE. 2024, pp. 1–5.

[19] Pranesh Kulkarni et al. “Plant Disease Detection Using Image Processing and Machine Learning”. In: *CoRR* abs/2106.10698 (2021). arXiv: 2106.10698. URL: https://arxiv.org/abs/2106.10698.

[20] Sue Han Lee et al. “How deep learning extracts and learns leaf features for plant classification”. In: *Pattern recognition* 71 (2017), pp. 1–13.

[21] Koay K Leong and Lim L Tze. “Plant leaf diseases identification using convolutional neural network with treatment handling system”. In: *2020 IEEE international conference on automatic control and intelligent systems (I2CACIS).* IEEE. 2020, pp. 39–44.

[22] Fatma Marzougui, Mohamed Elleuch, and Monji Kherallah. “A deep CNN approach for plant disease detection”. In: *2020 21st international arab conference on information technology (ACIT).* IEEE. 2020, pp. 1–6.

[23] Md Humaion Kabir Mehedi et al. “Plant leaf disease detection using transfer learning and explainable ai”. In: *2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).* IEEE. 2022, pp. 0166–0170.

[24] Sammy V Militante, Bobby D Gerardo, and Nanette V Dionisio. “Plant leaf detection and disease recognition using deep learning”. In: *2019 IEEE Eurasia conference on IOT, communication and engineering (ECICE).* IEEE. 2019, pp. 579–582.

[25] Faye Mohameth, Chen Bingcai, and Kane Amath Sada. “Plant disease detection with deep learning and feature extraction using plant village”. In: *Journal of Computer and Communications* 8.6 (2020), pp. 10–22.

[26] Nafisha Binte Moin et al. “Disease detection of bangladeshi crops using image processing and deep learning-a comparative analysis”. In: *2022 2nd International Conference on Intelligent Technologies (CONIT).* IEEE. 2022, pp. 1–8.

[27] J Arun Pandian et al. "Plant disease detection using deep convolutional neural network". In: *Applied Sciences* 12.14 (2022), p. 6982.

[28] Farhana Parvin. "Prevention is best defense against Bangladesh crop diseases, researchers say". In: *coRR* (2015).

[29] Lokesh R. *Rice Leaf Diseases Detection*. https://www.kaggle.com/datasets/loki4514/rice-leaf-diseases-detection. Accessed: 2025-06-17. 2023.

[30] M Hasinur Rahman1&2 and M Jashim Uddin. "Blast: A threat to cereal crops in Bangladesh". In: (2017).

[31] Shima Ramesh Maniyath et al. "Plant Disease Detection Using Machine Learning". In: Apr. 2018, pp. 41–45. DOI: 10.1109/ICDI3C.2018.00017.

[32] Muhammad Rizwan Saeed and Javed Rashid. *Potato Disease Leaf Dataset (PLD)*. https://www.kaggle.com/datasets/rizwan123456789/potato-disease-leaf-datasetpld. Accessed: 2025-06-17. 2021.

[33] Laboni Saha and R Lalmawipuii. "AI-Driven Solutions for Early Detection of Plant Diseases". In: *SHS Web of Conferences*. Vol. 216. EDP Sciences. 2025, p. 01076.

[34] Ramprasaath R Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626.

[35] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations (ICLR)* (2015). arXiv: 1409.1556. URL: https://arxiv.org/abs/1409.1556.

[36] Radhakrishnan Sujatha et al. "Performance of deep learning vs machine learning in plant leaf disease detection". In: *Microprocessors and Microsystems* 80 (2021), p. 103615.

[37] Divyansh Tiwari et al. "Potato leaf diseases detection using deep learning". In: *2020 4th international conference on intelligent computing and control systems (ICICCS)*. IEEE. 2020, pp. 461–466.

[38] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017. arXiv: 1706.03762. URL: https://arxiv.org/abs/1706.03762.

[39] Vibhor Kumar Vishnoi et al. "Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network". In: *IEEE Access* 11 (2023), pp. 6594–6609. DOI: 10.1109/ACCESS.2022.3232917.